



Università degli studi del Sannio

Dipartimento di ingegneria

Corso Dottorato di Ricerca in
“TECNOLOGIE DELL’INFORMAZIONE PER L’INGEGNERIA”
XXXII CICLO

LE RETI NEURALI ARTIFICIALI PER LA STIMA DEI FLUSSI DI TRAFFICO NELL’AMBITO DEI SISTEMI DI MONITORAGGIO

Relatore

Prof. Mariano Gallo

Coordinatore

Prof. Massimiliano Di Penta

Candidata

Giuseppina De Luca

D500300007

Introduzione	3
1. LE RETI NEURALI ARTIFICIALI.....	4
1.1 Generalità.....	4
1.2 Il processo di apprendimento	5
1.3 Capacità, overfitting e underfitting	11
1.4 Campi di applicazione	14
2 STATO DELL'ARTE.....	16
2.1 Le ANN e la loro applicazione alla previsione dei flussi di traffico.....	16
2.2 Applicazione delle ANN per la previsione a breve e lungo termine sui flussi di traffico	17
2.3 Applicazione delle ANN per l'estensione spaziale dei flussi di traffico	19
2.4 Contributo della tesi.....	20
3 USO DELLE RETI NEURALI ARTIFICIALI PER L'ESTENSIONE SPAZIALE DEI FLUSSI DI TRAFFICO SU RETE STRADALE URBANA	24
3.1 Introduzione	24
3.2 Sistemi e tecnologie di monitoraggio del traffico stradale	25
3.3 Approccio proposto.....	26
3.4 Sperimentazione e risultati.....	28
3.4.1 Test su intera rete stradale della città di Benevento	28
3.4.2 Test su singoli quartieri della città di Benevento	36
3.5 Conclusioni	41
4 RETI NEURALI ARTIFICIALI PER LA PREVISIONE DEI FLUSSI DI PASSEGGERI SU LINEE DI METROPOLITANA	45
4.1 Introduzione e articolazione del capitolo	45
4.2 Stato dell'arte	46
4.3 Descrizione del problema e approccio ANN	46
4.4 Generazione dei set di dati di addestramento	49
4.5 Caso di studio e risultati numerici	50

4.6	Conclusioni	54
5	CONCLUSIONI.....	57
6	LISTA PUBBLICAZIONI.....	59
6.1	Lista pubblicazioni tratte dalla tesi di Dottorato.....	59
6.2	Altre pubblicazioni	59

Introduzione

Il lavoro di tesi è focalizzato sull'applicazione delle Reti Neurali Artificiali (Artificial Neural Network, ANN) nell'ambito del monitoraggio dei flussi di mobilità. In particolare, lo scopo è valutare la capacità delle ANN di estendere spazialmente i dati rilevati in tempo reale da sensori o stazioni di monitoraggio. Le ANN sono state sperimentate sia nel caso del traffico stradale che nel caso di una rete ferroviaria urbana, su casi studio di dimensioni reali.

La tesi è articolata come segue.

Nel primo capitolo si introducono le reti neurali artificiali, fornendo una panoramica sul loro utilizzo nei diversi campi di applicazione quali finanza, riconoscimento ed elaborazioni delle immagini, diagnosi mediche, data mining, ingegneria dei trasporti, e così via. Inoltre, è illustrato il loro funzionamento generale e la loro struttura.

Nel secondo capitolo si esamina lo stato dell'arte sulle reti neurali artificiali applicate al settore dei trasporti, identificando il contributo innovativo della ricerca svolta.

Nel terzo capitolo si propone e si sperimenta l'applicazione delle ANN all'estensione spaziale dei flussi di traffico in una rete stradale urbana, sperimentando la metodologia proposta sulla rete della città di Benevento.

Nel quarto capitolo le reti neurali sono utilizzate per la stima dei flussi di utenti su una linea di metropolitana; in particolare, i flussi su alcune tratte non monitorate sono stimati in funzione di flussi rilevati su altre tratte.

Nel quinto capitolo sono tratte le conclusioni del lavoro e le prospettive di ricerca.

1. LE RETI NEURALI ARTIFICIALI

1.1 Generalità

Le Reti Neurali Artificiali (Artificial Neural Network - ANN) si basano su una analogia con il cervello umano. Questo è composto da un'intricata rete di connessioni di cellule in grado di influenzarsi fra loro che, lavorando insieme, cercano di realizzare un compito specifico. Queste cellule sono i *neuroni*, il cui schema di funzionamento è illustrato in figura 1.1., cellule nervose aventi dei prolungamenti chiamati *dendriti*, con molte ramificazioni, attraverso i quali il neurone riceve segnali elettrici da altri neuroni. Ogni neurone è formato anche da un prolungamento filamentoso detto *assone* che si ramifica all'estremità dei terminali attraverso i quali si propagano i segnali elettrici destinati alle altre cellule o ai dendriti di altre cellule.

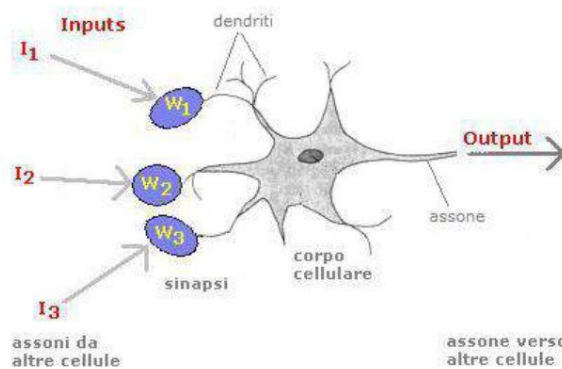


Figura 1.1 - Schema illustrativo del funzionamento di un neurone

La connessione fra un terminale ed un dendrite è chiamata *sinapsi*. In presenza di una differenza di potenziale fra esterno e interno della cellula, il neurone si attiva trasmettendo un impulso elettrico attraverso il suo assone che provoca la liberazione di un neurotrasmettitore.

Il cervello è un calcolatore molto complesso e potente e ha una capacità fondamentale, quella di imparare, cioè di modificare le connessioni fra i neuroni in base all'esperienza acquisita. Uno strumento di calcolo, come le reti neurali artificiali, che voglia riprodurre la capacità di elaborazione del cervello umano, deve essere realizzato attraverso una rete di elementi computazionalmente semplici, massicciamente paralleli, capace di apprendere e di riprodurre output in funzione di input noti, cioè di acquisire capacità di generalizzazione.

Per sviluppare una rete neurale efficiente bisogna, come per un cervello umano, farla apprendere.

1.2 Il processo di apprendimento

Il processo di apprendimento si basa su algoritmi di *machine learning* che permettono di allenare la rete per migliorarne l'accuratezza dei calcoli. Un neurone artificiale di una rete neurale è costituito in genere da molti ingressi e un'unica uscita. Al singolo ingresso è associato un peso che determina la conducibilità del canale di ingresso. L'uscita del neurone è rappresentata da una funzione della somma pesata degli ingressi. L'addestramento della rete neurale si basa sulla preparazione di una serie di esempi, di cui la rete conosce la natura, che costituiscono il dataset di training. Le risposte fornite dalla rete per i campioni di addestramento vengono confrontate con le risposte attese valutando l'errore fra le due. In base a questo errore i pesi della rete sono modificati. Questa procedura è ripetuta fino a quando le uscite della rete non forniscono un errore al di sotto di una soglia prestabilita (Bishop, 1995).

Dal punto di vista qualitativo una rete neurale è strutturata come segue. Dato un neurone avente n canali di ingresso $x_1, \dots, x_n$, ad ogni ingresso è associato un peso w_i , costituito da un numero reale, che riproduce le sinapsi neurali. Se il peso è positivo il canale si dice eccitatorio, se è negativo inibitorio. Il segnale di uscita è ricavato attraverso la funzione di attivazione, come somma pesata degli ingressi che è detta livello di attivazione:

$$a = \sum_{i=1}^n w_i x_i$$

la funzione di attivazione del neurone è espressa come:

$$f(a) = f\left(\sum_{i=1}^n w_i x_i\right) \quad (1.1)$$

dove f è la funzione di attivazione del neurone ed è anche chiamata funzione di trasferimento.

È possibile introdurre una soglia avente l'effetto di abbassare il valore di ingresso della funzione di trasferimento, in questo modo la (1.1) diventa:

$$y = f(a) = f\left(\sum_{i=0}^n w_i x_i - \theta\right)$$

La soglia può essere interpretata come il peso associato ad un ulteriore canale di ingresso x_0 detto *bias* di valore costante pari a 1. Quindi la funzione assume la forma della (1.1) con l'indice i che questa volta parte da 0.

La funzione di trasferimento (fig.1.2), definisce l'uscita di un neurone in funzione del suo livello di attivazione. Il segnale di uscita può essere un numero reale appartenente ad un dato intervallo.

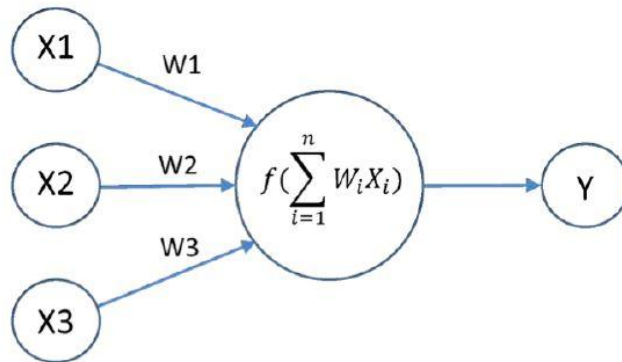


Figura 1.2 - Input e output di un neurone della rete. In entrata il neurone riceve i segnali da altri neuroni della rete pesati dai rispettivi w_i . In uscita il neurone fornisce la somma pesata attraverso la funzione f .

Le funzioni di attivazione sono componenti cruciali nelle reti neurali artificiali, in quanto determinano l'output di un neurone in base al suo input ponderato. Esse introducono la non linearità nel modello, consentendo alla rete di apprendere e modellare relazioni complesse. Diverse sono le funzioni che si utilizzano nelle reti neurali; una scelta molto comune per la funzione di trasferimento è la sigmoide:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Un'altra funzione di attivazione importante è la *ReLU* (*rectified liner unit*):

$$f(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$$

Questa funzione assumerà un ruolo importante per gli algoritmi di *deep learning* per la sua rapidità di apprendimento per reti formate da molti strati di neuroni.

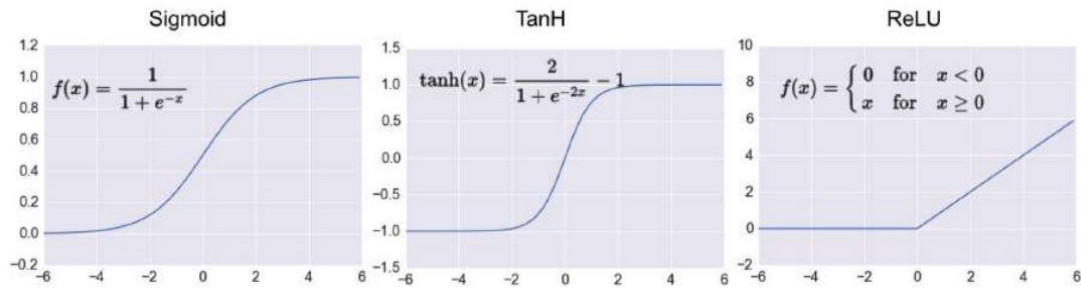


Figura 1.3 - Grafici di tre funzioni di attivazione. Rispettivamente sigmoide, tangente iperbolica e ReLU.

La scelta della funzione di attivazione (fig.1.3) dipende dal problema specifico che la rete neurale deve risolvere. Sperimentare con diverse funzioni di attivazione durante l'addestramento può aiutare a ottimizzare le prestazioni della rete.

Prima di analizzare la metodologia di addestramento delle reti neurali, bisogna introdurre alcune definizioni importanti; verranno, poi, descritte sommariamente alcune tipologie di reti come esempio illustrativo.

Una rete neurale si dice stratificata quando è formata da vari strati di neuroni, tali che ognuno di loro sia connesso con quelli dello strato successivo, ma senza che vi siano connessioni tra neuroni dello stesso strato. In questo tipo di rete non avviene nessuna elaborazione nello stato di ingresso. Una rete in cui il segnale di ingresso viaggia sempre in avanti dall'ingresso all'uscita si chiama *feedforward*. In generale in questo tipo di reti ci possono essere uno o più strati nascosti (*hidden layers*), strati di neuroni che non comunicano direttamente con l'esterno. Questi strati facilitano il compito della rete permettendole di rappresentare in maniera opportuna gli stimoli di ingresso. Nella Figura 1.4 si riporta una rappresentazione grafica di una rete stratificata.

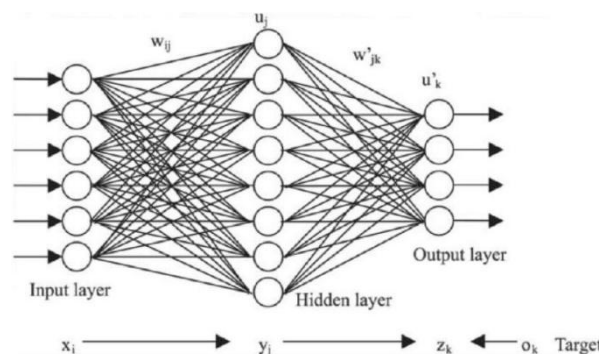


Figura 1.4 - Esempio di rappresentazione grafica di una rete stratificata avente uno strato di unità nascoste.

Nelle reti neurali, tipicamente, si assume che ogni input (ed ogni output) sia indipendente da tutti gli altri; per questo motivo le ANN risultano utili nei problemi di previsione (come quelli del completamento dei testi). Eseguendo le stesse elaborazioni per tutti gli elementi di una sequenza di input, l'output dipende anche da tutte le elaborazioni precedenti. Queste reti assumono un ruolo importante nei campi del riconoscimento vocale e ortografico.

La sfida principale nella *machine learning* è sicuramente tarare gli algoritmi di apprendimento affinché forniscano buone prestazioni se applicati su dati di input sconosciuti. Al fine di avere dei risultati soddisfacenti bisogna addestrare la rete nello svolgimento del compito assegnato.

Come è stato già detto in precedenza, il punto di partenza per l'addestramento è avere a disposizione un dataset formato da un "insieme di addestramento" che verrà somministrato in input alla rete, per allenarla, e un "insieme di test" per verificare la validità e la qualità dell'addestramento stesso. Durante l'addestramento, dopo aver fornito i dati dell'insieme di addestramento in input, la rete produce un output in forma di vettore di risultati. È necessario calcolare una funzione che misuri l'errore (o la distanza) tra i risultati di output e i pattern desiderati dei risultati, cioè i valori con cui vengono etichettati i dati nell'apprendimento supervisionato. Questa funzione viene chiamata di solito funzione di costo, funzione di perdita o funzione errore. La rete, in fase di addestramento modifica i suoi parametri interni in modo da minimizzare questa funzione. I parametri sono i pesi e, spesso, sono numeri reali che possono essere visti come delle "manopole" che definiscono la funzione di input-output della rete. Per modificare in maniera appropriata il vettore dei pesi, l'algoritmo di apprendimento calcola un vettore gradiente che, per ogni peso, indica di quanto l'errore cresca o decresca se i pesi sono incrementati di una quantità infinitesima. La funzione obiettivo, mediata su tutti gli esempi di addestramento, può essere rappresentata nello spazio multi-dimensionale del vettore dei pesi, come mostrato nell'esempio di Figura 1.5.

Il vettore gradiente indica la direzione più ripida di discesa in ogni step, avvicinandosi sempre di più al minimo, dove l'errore di output è mediamente inferiore.

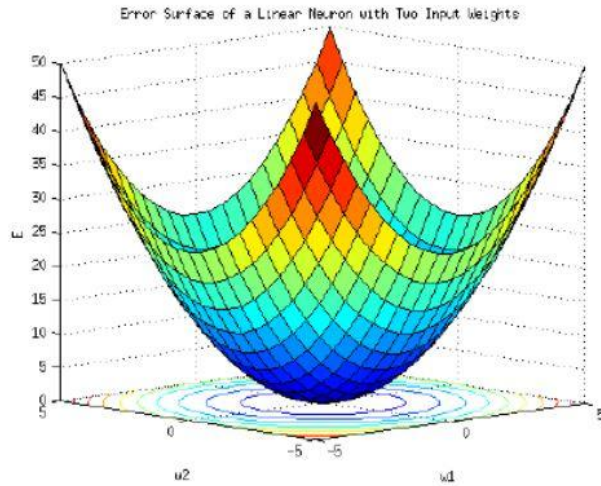


Figura 1.5 – Esempio di funzione obiettivo

L'approccio più utilizzato per minimizzare l'errore è l'algoritmo di gradiente stocastico decrescente (SGD) (Rotondi et al., 2006). Esso prevede di mandare in input alcuni esempi, calcolando l'output, l'errore associato e il gradiente medio variando i pesi. La procedura è iterata per molti set piccoli di esempi presi dall'insieme di addestramento finché la media della funzione obiettivo non smette di decrescere. Si parla di gradiente stocastico perché ogni piccolo insieme di esempi fornisce una stima del gradiente medio su tutti gli altri. Dopo questa procedura si passa alle valutazioni sulla performance della rete; essa viene misurata con gli esempi dell'insieme di test. Questo serve a testare la capacità di generalizzazione della rete. A titolo di esempio, per descrivere il processo di addestramento, supponiamo di voler costruire una rete neurale artificiale stratificata di tipo feedforward con n_{var} neuroni di ingresso, con uno strato di neuroni nascosti n_b e con un unico neurone in uscita. Il neurone in uscita avrà una funzione di attivazione data dalla somma lineare delle funzioni d'uscita dei neuroni dello strato nascosto:

$$y_{ANN} = \sum_{j=1}^{n_b} f \left(\sum_{i=1}^{n_{var}} w_{ij} x_i \right) w_{ji}$$

dove w_{ij} è il peso tra il neurone di ingresso i il neurone j dello strato nascosto, mentre w_{ji} è il peso tra il neurone j dello strato nascosto ed il neurone di uscita della rete.

La rete sia addestrata con un campione contenente N eventi:

$$x_m = (x_1 \dots x_{n_{var}})m$$

Con $m = 1, \dots, N$.

Per ogni evento di addestramento m , la rete neurale calcola la funzione di uscita $y_{ANN, m}$ e la confronta con il valore di uscita atteso y_m .

La funzione di errore, che misura lo scostamento fra l'uscita della rete neurale ed il valore di uscita atteso, è definita da:

$$E(x_1, \dots, x_N | \mathbf{w}) = \sum_{m=1}^N E_m(x_m | \mathbf{w}) = \sum_{m=1}^N \frac{1}{2} (y_{ANN, m} - y_m)^2$$

dove $\mathbf{w}$ denota l'insieme dei pesi della rete neurale. Il migliore insieme di pesi possibili è quello che minimizza la funzione di errore. Questo set di pesi può essere calcolato attraverso il metodo del gradiente decrescente (SGD): partendo da un insieme di pesi casuali $\mathbf{w}_p$, i pesi della rete sono variati attraverso piccoli spostamenti nello spazio a w dimensioni nelle direzioni $\nabla_{\mathbf{w}} E$ dove E decresce molto rapidamente:

$$\mathbf{w}^{\rho+1} = \mathbf{w}^{(\rho)} - \eta \nabla_{\mathbf{w}} E$$

dove η indica un numero positivo compreso tra 1 e 0 che rappresenta il tasso di apprendimento, mentre con ρ si indica lo strato di connessioni a cui i pesi fanno riferimento.

I pesi di connessione fra lo strato di neuroni nascosti ed il neurone di uscita vengono aggiornati attraverso la seguente relazione:

$$\Delta w_{ji} = -\eta \sum_{m=1}^N \frac{\partial E_m}{\partial w_{ji}} = -\eta \sum_{m=1}^N (y_{ANN, m} - y_m) y_{j, m} \quad (1.11)$$

Gli algoritmi i cui i pesi vengono aggiornati con la (1.11) si dicono di back-propagation, i pesi sono calcolati partendo dall'ultimo strato di neuroni procedendo all'indietro. Un ciclo di presentazione alla rete di tutti gli eventi appartenenti all'insieme di addestramento è detto *epoca*.

Dopo l'addestramento della rete, si valutano le sue prestazioni con il test set. Gli eventi appartenenti al test set non devono essere stati analizzati dalla rete.

In sintesi, un algoritmo di apprendimento può essere creato unendo la combinazione specifica di una serie di dataset, una funzione di costo, una procedura di ottimizzazione e un modello. Si può ottenere un vasto range di algoritmi indipendenti tra loro cambiando alcuni di questi componenti.

La funzione di costo include tipicamente almeno un termine che causi il processo di apprendimento per preformare al meglio la stima statistica dei dati. Essa può contenere

anche termini aggiuntivi, per esempio termini di regolarizzazione. In caso di modelli non lineari la maggior parte delle funzioni di costo non possono essere ottimizzate con forme chiuse; bisogna, in questi casi, introdurre un metodo numerico iterativo di ottimizzazione, come quello del gradiente prima descritto.

1.3 Capacità, overfitting e underfitting

L'abilità di un algoritmo di performare bene su input mai osservati precedentemente si chiama *generalizzazione*. Come già detto in precedenza, durante la fase di addestramento, si testa l'efficienza della rete mandando in input dei dati di test, in modo da calcolarne l'errore associato. La performance dell'algoritmo viene migliorata cercando di minimizzare questo errore.

Questo è un semplice problema di ottimizzazione. Ciò che separa il machine learning dall'ottimizzazione è la minimizzazione non dell'errore di addestramento, ma dell'errore sulla generalizzazione. L'errore di generalizzazione è definito come il valore atteso dell'errore su un nuovo input. Il valore atteso è valutato su una serie di input differenti, estratti dalla distribuzione di input che ci aspettiamo il sistema incontri nella pratica. Risulta spontanea la domanda su come si possa influenzare l'errore sull'insieme di test quando questo non si conosce a priori. Il campo della teoria dell'apprendimento statistico fornisce alcune risposte. Attraverso delle ipotesi riguardanti la maniera in cui sono stati raccolti i dati si possono studiare matematicamente le relazioni che intercorrono tra l'errore dell'insieme di addestramento e l'errore sull'insieme del test.

Assumiamo che entrambi gli insiemi siano campionati casualmente seguendo una distribuzione di probabilità sull'insieme globale dei dati (processo di generazione dei dati), e che gli esempi in ogni dataset siano indipendenti tra di loro. Questa distribuzione condivisa viene chiamata distribuzione della generazione dei dati, che denotiamo con p_{data} .

Si può osservare subito che l'errore sull'insieme di addestramento è uguale al valore atteso dell'errore sull'insieme di test dello stesso modello. Questo perché i due valori attesi sono stati trovati usando la stessa procedura di generazione dall'insieme di tutti i dati. L'unica differenza sulle due condizioni è il nome che assegniamo al dataset del campione ("addestramento" e "test").

Quindi, dopo aver generato l'insieme di addestramento, si scelgono e si variano i parametri per ridurre l'errore sull'insieme, che corrisponde anche all'errore sull'insieme di test. Successivamente viene generato l'insieme di test.

I fattori che determinano quanto sia efficiente un algoritmo di apprendimento sono:

- la sua abilità nel minimizzare l'errore di addestramento;
- la sua abilità nel minimizzare il gap tra errore di addestramento e l'errore di test.

Questi due fattori corrispondono alle due sfide principali del *machine learning*: *l'underfitting* e *l'overfitting*. L'*underfitting* si verifica quando il modello non è capace di ottenere un errore sufficientemente piccolo per l'insieme di addestramento. L'*overfitting* si verifica quando il gap tra l'errore sull'insieme di addestramento e l'errore sull'insieme di test è troppo grande. Si può controllare se un modello tende di più all'*overfitting* o all'*underfitting* alterando la sua capacità. La capacità di un modello è la sua abilità nell'approssimare una larga varietà di funzioni. Modelli con alta capacità potrebbero presentare *overfitting* adattandosi troppo alle caratteristiche dell'insieme di training non generalizzando sull'insieme di test.

Una maniera per controllare la capacità di un algoritmo di apprendimento è la scelta del suo spazio delle ipotesi, cioè, il set di funzioni che l'algoritmo è capace di attribuire come soluzione del problema. Per esempio, l'algoritmo di regressione lineare possiede come spazio delle ipotesi un insieme di tutte le funzioni lineari che verranno fornite in input. Possiamo generalizzare la regressione lineare includendo nello spazio delle ipotesi le funzioni polinomiali, piuttosto che solo quelle lineari. In questa maniera aumenterà la capacità del modello.

Generalmente gli algoritmi di *machine learning* hanno delle performance migliori quando la capacità dei dati è appropriata per la reale complessità del compito che devono svolgere e per la quantità di dati dell'insieme di addestramento forniti. Modelli con capacità insufficiente non sono capaci di svolgere compiti complessi. Modelli con grande capacità possono risolvere compiti difficili, ma nel caso in cui la capacità sia più grande di quella richiesta dal compito da svolgere potrebbero andare in *overfitting*.

Ci sono diversi modi per variare la capacità del modello. Essa non è determinata solo dalla scelta del modello. Il modello specifica quale famiglia di funzioni dell'algoritmo di apprendimento può scegliere quando deve variare i parametri in maniera da ottimizzare l'addestramento. In questo caso si parla di capacità effettiva. In molti casi questa situazione corrisponde ad un problema di ottimizzazione. In pratica, l'algoritmo non sceglie realmente la funzione migliore, ma quella che riduce significativamente l'errore

dell'insieme di addestramento. Queste ulteriori limitazioni, come l'imperfezione dell'ottimizzazione dell'algoritmo, potrebbero rendere la capacità effettiva minore della capacità di rappresentazione della famiglia di modelli.

La teoria di apprendimento statistico fornisce varie definizioni quantitative della capacità del modello. Una di queste è la dimensione di Vapnik-Chervonenkis (VC). La dimensione VC misura la capacità di un classificatore lineare. È definita come il più grande valore di m che il classificatore può etichettare arbitrariamente su un insieme di addestramento formato da m punti. Quantificare la capacità del modello permette di fare predizioni quantitative. Il risultato più importante nella teoria statistica dell'apprendimento è che la discrepanza tra errore di training ed errore di generalizzazione aumenta all'aumentare della capacità del modello e diminuisce all'aumentare del numero di esempi disponibili di training. Si ricorda che, mentre le funzioni più semplici sono più utili per la generalizzazione, siccome hanno un gap più piccolo tra errore di training ed errore di test, bisogna scegliere contemporaneamente un numero sufficientemente complesso di ipotesi per minimizzare l'errore di training. Spesso l'errore decresce finché non raggiunge asintoticamente il valore minimo dell'errore corrispondente al punto in cui la capacità del modello aumenta.

Tipicamente l'errore di generalizzazione ha la forma di una curva ad U in funzione della capacità del modello. Nella Figura 1.6 è rappresentato l'errore di addestramento e di generalizzazione in funzione della capacità dei dati.

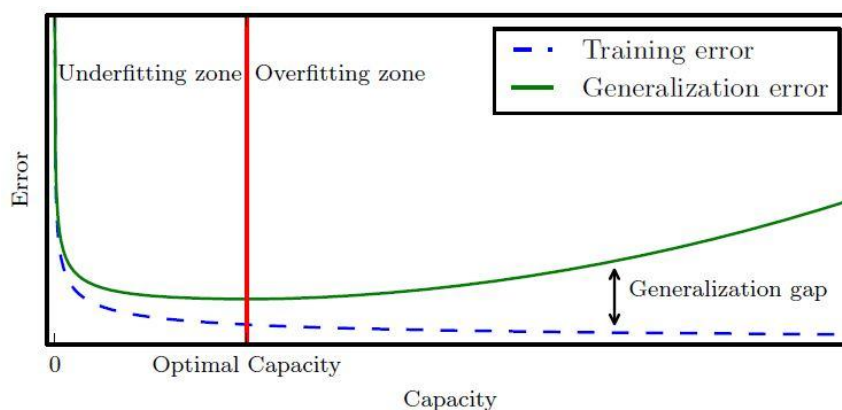


Figura 1.6 - Errore di addestramento e di generalizzazione in funzione della capacità dei dati. Con l'aumentare di quest'ultima aumenta il gap tra i due errori.

1.4 Campi di applicazione

Le reti neurali artificiali sono utilizzate in molti settori ed hanno numerosi campi di applicazione.

In *medicina*, per la diagnosi di malattie tramite l'analisi di immagini mediche, la previsione di risultati clinici e la scoperta di nuovi farmaci attraverso il data mining di grandi dataset.

Nel settore *finanziario*, le reti neurali sono utilizzate per il rilevamento di frodi, la previsione di tendenze di mercato e la gestione del rischio. Algoritmi basati su reti neurali sono in grado di analizzare grandi volumi di dati finanziari in tempo reale.

Le reti neurali svolgono un ruolo cruciale anche nella robotica, consentendo ai robot di apprendere e eseguire compiti complessi attraverso l'esperienza. Ciò include la manipolazione di oggetti, la navigazione autonoma e la collaborazione con esseri umani.

Nel campo *dell'ingegneria dei trasporti* le reti neurali possono essere utilizzate in diversi ambiti:

1. Monitoraggio del Traffico e Previsione dei Flussi

Analizzando i dati provenienti da sensori stradali, telecamere di sorveglianza e altri dispositivi di monitoraggio per valutare la situazione del traffico in tempo reale. Inoltre, possono essere utilizzate per prevedere i flussi di traffico futuri, aiutando nella pianificazione e nella gestione del traffico.

2. Sistemi di Controllo del Traffico Intelligenti

L'integrazione di reti neurali nei sistemi di controllo del traffico consente di ottimizzare la gestione dei semafori, regolando i tempi di verde, rosso e giallo in tempo reale in base alle condizioni effettive del traffico. Ciò contribuisce a ridurre la congestione stradale e migliorare la fluidità del traffico.

3. Navigazione Ottimizzata e Pianificazione del Percorso

Sistemi di navigazione basati su reti neurali possono offrire percorsi ottimizzati tenendo conto delle condizioni del traffico in tempo reale. Ciò non solo riduce il tempo di percorrenza per gli automobilisti, ma contribuisce anche a distribuire il traffico in modo più uniforme.

4. Manutenzione Predittiva delle Infrastrutture

Le Reti Neurali possono essere utilizzate per prevedere la necessità di manutenzione stradale predittiva. Analizzando dati relativi alle condizioni delle strade e dei ponti, è

possibile identificare potenziali problemi prima che si verifichino, contribuendo a garantire la sicurezza e prolungare la durata delle infrastrutture.

5. Gestione dei Parcheggi

Sistemi di gestione del parcheggio basati su reti neurali possono aiutare a monitorare e ottimizzare l'uso degli spazi di parcheggio. Le reti neurali possono essere utilizzate per rilevare automaticamente la disponibilità di posti auto e guidare gli automobilisti verso le aree con posti liberi.

6. Analisi di Incidenti e Sicurezza Stradale

Le Reti Neurali possono analizzare dati storici sugli incidenti stradali per identificare modelli e fattori di rischio. Queste analisi possono guidare gli sforzi per migliorare la sicurezza stradale, ad esempio attraverso l'implementazione di misure preventive o la modifica delle infrastrutture.

Bibliografia Capitolo 1

Bishop, C.M.; Neural networks for pattern recognition. Oxford university press, 1995.

Rotondi, A.; Pedroni, P.; Pievatolo, P.; Probabilità statistica e simulazione: programmi applicativi scritti con Scilab. Springer Science & Business Media, 2006.

2 STATO DELL'ARTE

2.1 Le ANN e la loro applicazione alla previsione dei flussi di traffico

La Rete Neurale Artificiale (ANN), come introdotto nel capitolo precedente, è un approccio ampiamente diffuso per prevedere o riprodurre diversi fenomeni in diversi settori scientifici. I modelli implementati con le ANN sono non parametrici: non ci sono parametri espliciti che collegano i dati di input e di output, ma le relazioni tra questi dati sono nascoste e non interpretabili. Per questi motivi, le ANN sono ampiamente utilizzate per prevedere come variano alcuni dati (di output) in funzione di altri dati (di input), se non è necessario conoscere esplicitamente le relazioni tra i dati o se è molto difficile identificare tali relazioni.

La non parametricità delle ANN è al tempo stesso il punto di forza e di debolezza dell'approccio. Infatti, questi modelli, se ben addestrati, sono in grado di dare buoni risultati in tempi molto brevi, sono in grado di riprodurre problemi molto complessi per i quali un approccio parametrico è difficile da implementare e possono essere calibrati (addestrati) senza comprendere e identificare i parametri del problema. D'altra parte, i modelli calibrati (addestrati) non sono esportabili ad altri casi di studio simili e sono validi solo se le condizioni al contorno non cambiano in modo significativo.

Da queste considerazioni preliminari, risulta evidente come le ANN possano essere utilmente adottate per la previsione dei flussi di utenti sulle reti di trasporto; infatti, in alcuni casi, come ad esempio i problemi di controllo, è fondamentale stimare rapidamente i flussi di traffico e le relazioni tra i flussi non devono essere necessariamente esplicitate. In questo capitolo si analizzano i principali lavori che trattano l'uso delle ANN nel settore dei trasporti si individuano le potenzialità delle stesse nella previsione del traffico, sia temporale (previsioni a breve termine) sia spaziale (estensione dei dati di traffico da arco ad arco) e si riportano le altre possibili applicazioni nel settore dei trasporti.

Una rassegna della letteratura sulle ANN applicate al settore dei trasporti era già stata proposta in Gallo et al. (2016); in questo lavoro si studiano principalmente i metodi di previsione dei flussi di utenti.

I principali lavori nel campo dei trasporti pubblicati prima del 1995 sono stati esaminati da Dougherty (1995) che ha identificato 13 aree tematiche principali:

- comportamento del conducente/veicolo autonomo;
- stima dei parametri;
- manutenzione delle pavimentazioni;

- rilevamento/classificazione dei veicoli;
- analisi dei modelli di traffico;
- gestione del trasporto merci;
- previsione del traffico;
- politica ed economia dei trasporti;
- trasporto aereo;
- trasporto marittimo;
- veicoli sottomarini;
- esercizio delle metropolitane;
- controllo del traffico.

Una rassegna della letteratura sulle ANN applicate alla previsione dei flussi di traffico è riportata in De Luca e Gallo (2017).

Più di recente, le ANN e le loro varianti sono state utilizzate per il riconoscimento dei segnali stradali, un compito essenziale in molte applicazioni come la sorveglianza del traffico e la guida autonoma; in questo contesto, le reti neurali profonde sono state proposte da Ciresan et. al. (2012) e Arcos-Garcia et. al. (2018). Applicazioni delle ANN al problema del minimo percorso si trovano in Sang et al. (2016) e Huang et al. (2017), mentre il rilevamento automatico degli incidenti stradali è stato studiato da Srinivasan et al. (2008). Tanprasert et al. (2017) hanno proposto di utilizzare le ANN per l'identificazione dei conducenti.

2.2 Applicazione delle ANN per la previsione a breve e lungo termine sui flussi di traffico

Per quanto riguarda le previsioni di traffico, in letteratura si trovano numerosi articoli; due rassegne riguardano la previsione a breve termine (Vlahogianni, et al., 2014; Oh et al., 2015), che è stata ampiamente studiata con le ANN. In effetti, si tratta di un problema per il quale l'approccio a scatola nera non costituisce un ostacolo: non abbiamo bisogno di esportare il modello ad altri casi o di conoscere esplicitamente le relazioni tra input e output. L'applicabilità delle ANN a tali problemi è stata studiata da Kirby et al. (1997), mentre numerosi studi si sono concentrati sulle applicazioni delle previsioni a breve termine alle autostrade (Florio e Mussone, 1995), ai contesti urbani (Zheng e Van Zuylen 2013; Gao e Sun, 2010; Gao et al., 2013; Ledoux, 1997) o alle superstrade urbane (Li and Lu, 2009). I lavori di Lin et al. (2008) e Zheng Zhu et al. (2014) hanno considerato, oltre alla previsione a breve termine, un'estensione spaziale dei dati.

Un altro argomento riguarda la stima della matrice origine-destinazione, per la quale si rimanda a Yang et al. (1992) e Mussone et al. (2010).

L'analisi della letteratura si concentra principalmente sugli studi che possono essere utilizzati per stimare/prevedere i flussi di traffico sulle reti stradali e i flussi di passeggeri sulle reti di trasporto pubblico.

Classifichiamo gli studi nelle seguenti aree tematiche:

- previsione dei flussi di traffico/utenti a breve termine;
- estensione spaziale dei flussi di traffico/utenti;
- stima della matrice origine-destinazione.

La previsione a breve termine è la più ricca area di applicazione delle ANN nel settore dei trasporti, presenta un numero significativo di applicazioni nel mondo reale e potrebbe essere utilizzata per gestire meglio la congestione. Il problema è quello di prevedere i flussi di traffico negli intervalli di tempo successivi in funzione dei flussi di traffico rilevati negli intervalli di tempo precedenti. La letteratura sulla previsione del traffico a breve termine è riportata nei lavori di Vlahogianni et al. (2014). Si tratta del "classico" problema che può essere efficacemente risolto dalle ANN; infatti, i punti deboli delle ANN (necessità di esportare ad altri casi simili e stabilità delle condizioni al contorno) non costituiscono un problema e l'approccio non parametrico è assolutamente accettabile (non è utile collegare input e output con relazioni esplicite). Di seguito, vengono riassunti alcuni lavori che si sono concentrati sul traffico a breve termine.

Uno dei primi lavori a proporre le ANN per la previsione dei flussi di traffico a breve termine è stato pubblicato da Smith e Demetsky (1994), che hanno proposto di utilizzare le ANN per risolvere il problema e hanno confrontato i risultati ottenuti con altri approcci classici, sottolineando le potenzialità del metodo. Zhang et al. (1997) hanno utilizzato le ANN per modellare il traffico autostradale in un ambiente dinamico macroscopico. Dougherty e Kirby (1998) hanno utilizzato le ANN per prevedere la congestione del traffico. Park et al. (1998) hanno proposto una funzione ANN a base radiale per la previsione a breve termine.

Zheng et. al. (2006) hanno proposto un approccio di rete neurale combinata con approccio Bayesano per la previsione dei flussi di traffico sulle autostrade; in questo lavoro sono stati adottati intervalli di tempo di 15 minuti e il modello è stato testato su un'autostrada a Singapore.

Hodge et. al. (2011) hanno utilizzato reti neurali binarie per prevedere le condizioni del traffico in tre località urbane del Regno Unito. Kumar et. al. (2013) hanno sviluppato una

rete neurale con un'unica uscita per prevedere il volume di traffico in un singolo sito a partire da input che descrivono 19 diverse caratteristiche nei 45 minuti precedenti; queste caratteristiche includevano: giorno della settimana, ora del giorno, flusso di traffico classificato per tipologie di veicoli, velocità medie e densità complessiva del traffico. Questa architettura ha dato risultati promettenti per la previsione dei flussi di traffico nei 15 minuti successivi.

Zheng e Van Zuylen (2013) hanno proposto di utilizzare i veicoli sonda come sensori per raccogliere dati sul traffico in una rete urbana. Per stimare i tempi di percorrenza sull'intero percorso è stata utilizzata una ANN a tre strati. Gli autori hanno concluso che il modello ANN fornisce risultati migliori rispetto al modello analitico.

Nel lavoro di Csikos et al. (2015) è stato proposto un metodo di previsione della velocità del traffico basato su ANN per le reti stradali urbane. Park e Rilett (1999) hanno proposto l'uso di una rete neurale artificiale per la previsione dei tempi di percorrenza dei collegamenti autostradali. Altri studi incentrati sulla previsione dei flussi di traffico a breve termine mediante ANN si trovano nei lavori di Yasdi (1999), Li e Lu (2009), Gao e Sun (2010), Gao et. al. (2013), e Goves (2015).

Florio e Mussone (1995) hanno applicato un modello di rete neurale per valutare la relazione flusso-densità su un'autostrada, al fine di prevedere la stabilità del flusso di traffico.

Le ANN sono state utilizzate anche per stimare la matrice origine-destinazione in diversi lavori. Gong (1998) ha proposto un metodo basato su ANN per stimare i flussi origine-destinazione. Approcci basati su reti neurali artificiali per stimare le matrici OD in funzione dei conteggi di traffico sono stati proposti da Remya e Samson (2013) e De Martinis (2008, 2010). In tutti i casi i risultati sottolineano come le ANN siano in grado di contribuire a risolvere il problema, spesso superando in precisione altri metodi.

Questa analisi ha evidenziato che la previsione a breve termine è un tema consolidato nell'uso delle ANN; le altre due aree tematiche sono, invece, meno consolidate e necessitano di altre nuove analisi, in particolare per quanto riguarda l'estensione spaziale.

2.3 Applicazione delle ANN per l'estensione spaziale dei flussi di traffico

A differenza della previsione a breve termine, l'estensione spaziale dei flussi di traffico con le ANN, come si evince dall'analisi della letteratura, è stata meno studiata, anche se i primi risultati (Yang et. al., 1992) sono promettenti.

I lavori di letteratura hanno considerato solo i flussi di traffico stradale come variabili nell'ambito di sistemi di monitoraggio in tempo reale della rete stradale; l'estensione e le possibili applicazioni ai sistemi/reti di trasporto pubblico non sono state studiate allo stesso modo. Per quanto riguarda la stima della matrice OD, le ANN sono state studiate considerando principalmente i flussi rilevati come dati di input; in questo campo, alcuni risultati sembrano essere promettenti.

I limiti delle ANN, legati alla natura non parametrica dell'approccio, non permettono di applicare il modello a contesti diversi da quelli per cui la ANN è stata addestrata. Questo non è un problema per l'estensione spaziale se le condizioni al contorno non cambiano significativamente; se invece, vi sono forti perturbazioni alle caratteristiche del flusso (ad esempio un incidente), i risultati ottenuti, in particolare per le previsioni a breve termine, possono essere molto distanti dalla realtà o non ammissibili (ad esempio flussi negativi).

2.4 Contributo della tesi

A valle dell'analisi riportata in questo capitolo sono stati identificati due campi di ricerca da approfondire, in quanto non trattati compiutamente in letteratura, che sono oggetto del lavoro di tesi:

- 1) l'estensione spaziale dei dati sui flussi di autoveicoli rilevati da sensori;
- 2) la stima dei flussi di utenti su alcune tratte di una linea metropolitana in funzione di rilievi effettuati su altre tratte.

In entrambi i casi si applicheranno le ANN e si effettueranno dei test su casi di studio di dimensione reale.

Bibliografia Capitolo 2

Arcos-Garcia, A.; Alvares-Garcia, J.A.; Soria-Morillo, L.M.; Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural Networks*, vol. 99, pp. 158-165, 2018.

Ciresan, D.; Meier, U.; Masci, J.; Schmidhuber, J.; Multi-column deep neural network for traffic sign classification, *Neural Networks*, vol. 32, pp. 333-338, 2012.

Csikos, A.; Viharos, Z.J.; Kisk, B.; Tettamanti, T.; Varga, I.; Traffic speed prediction method for urban networks an ANN approach, *Models and Technologies for Intelligent Transportation Systems (MT-ITS)* Budapest, Hungary, 2015.

De Luca, G.; Gallo M.; Artificial Neural Networks for forecasting user flows in transportation networks: literature review, limits, potentialities and open challenges. Proceedings of 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, pp. 919-923, 2017.

De Martinis, V.; A Neural Network Approach to Motorway OD Matrix Estimation from Loop Counts, Journal of Transportation Systems Engineering and Information Technology, vol. 10, pp. 88-98, 2010.

De Martinis, V.; Una nuova metodologia d'uso delle reti neurali artificiali per l'aggiornamento della domanda di trasporto. Tesi di dottorato, Università di Napoli "Federico II", 2008.

Dougherty, M.; A review of neural networks applied to transport, Transportation Research Part C, vol. 3, pp. 247-260, 1995.

Dougherty, M.S.; Kirby, H.C.; The use of neural networks to recognize and predict traffic congestion, Traffic Eng. Control, vol. 346, pp. 311-314, 1998.

Florio, L.; Mussone, L.; Neural network models for classification and forecasting of freeway traffic flow stability. In Transportation Systems: Theory and Application of advanced technology; Liu B., Blosseville J.M. Eds., pp. 773-784, 1995.

Florio, L.; Mussone, L.; Neural network models for classification and forecasting of freeway traffic flow stability, in Transportation Systems: Theory and Application of advanced technology, Liu B., Blosseville J.M. (eds). pp. 773-784, 1995.

Gallo, M.; Simonelli, F.; De Luca, G.; Della Porta, C.; An Artificial Neural Network approach for spatially extending road traffic monitoring measures, Proceedings of 2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS 2016), Bari, Italy, pp. 209-213, June 2016.

Gao, J.; Leng, Z.; Qin, Y.; Ma, Z.; Liu, X.; Short-term traffic flow forecasting model based on wavelet neural network, 25th Chinese Control and Decision Conference, pp. 5081-5084, 2013.

Gao, Y.; Sun, S.; Multi-link traffic flow forecasting using neural networks, 2010 Sixth International Conference on Natural Computation, pp. 398-401, 2010.

Gong, Z.; Estimating the urban OD matrix: A neural network approach. European Journal of Operational Research, vol. 106, pp. 108-115, 1998.

Goves, C.; Short term traffic prediction on the UK motorway network using neural networks, Proceedings of European Transport Conference, 2015.

Hodge,V.; Austin, J.; Krishnan,R.; Polak,J.; Jackson, T.; Short-term traffic prediction using a binary neural network, UTSG, 2011.

Huang, W.; Yan, C.; Wang, J.; Wang, W.; A time-delay neural network for solving time-dependent shortest path problem. *Neural Networks*, vol. 90, pp. 21-28, 2017.

Kikuchi,S.; Nanda,R.; Perincherry, V.; A method to estimate trip O-D patterns using neural network approach, *Transportation Planning and Technology*, vol. 17, pp. 51-65, 1993.

Kirby, H.R.; Watson, S.M.; Dougherty, S.; Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International Journal of Forecasting*, vol. 13, pp. 43-50, 1997.

Kumar,K.; Parida,M.; Katiyar,V.K.; Short term traffic flow prediction for a non urban highway using Artificial Neural Network, *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 755-764, 2013.

Ledoux, C.; An urban traffic flow model integrating neural networks. *Transportation Research Part C*, vol. 5, pp. 287-300, 1997.

Li, R.; Lu, H.; Combined neural network approach for short-term urban freeway traffic flow prediction. *Lecture Notes in Computer Science*, vol. 5553, pp. 1017-1025, 2009.

Lin, S.; Xi, Y.; Yang, Y.; Short-Term Traffic Flow Forecasting Using Macroscopic Urban Traffic Network Model. *Atti della Conferenza internazionale IEEE sui sistemi di trasporto intelligenti, ITSC*, 2008.

Mussone, L.; Grant-Muller, S.; Chen, H.; A Neural Network Approach to Motorway OD Matrix Estimation from Loop Counts. *Journal of Transportation Systems Engineering and Information Technology*, vol. 10, pp. 88-98, 2010.

Oh, S.; Byon, Y.J.; Jang, K.; Yeo, H.; Short-term travel-time prediction on highway: a review of the data-driven approach. *Transport Reviews*, vol. 35, pp. 4-32, 2015.

Park, B.; Carroll, J.M.; Urbank, T.I.; Short-term freeway traffic volume forecasting using radial basis function neural network. *Transportation Research Record*, vol. 1651, pp. 39-46, 1998.

Park, D.; Rilett, L.R.; Forecasting Freeway Link Travel Times with a Multilayer Feedforward Neural Network, *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, pp. 357-367, 1999.

Remya, K.P.; Samson, M.; OD Matrix Estimation from Link Counts Using Artificial Neural Network, *International Journal of Scientific & Engineering Research*, vol. 4, 2013.

Sang, Y.; Lv, J.; Qu, H.; Yi, Z.; Shortest path computation using pulse-coupled neural networks with restricted autowave. *Knowledge-Based Systems*, vol. 114, pp. 1-11, 2016.

Smith, B.L.; Demetsky, M.J.; Short-term traffic flow prediction: Neural network approach, *Transportation Research Record*, vol. 1453, pp. 98–104, 1994.

Srinivasan, D.; Sharma, V.; Toh, K.A.; Reduced multivariate polynomial-based neural network for automated traffic incident detection. *Neural Networks*, vol. 21, pp. 484-492, 2008.

Tanprasert, T.; Saiprasert, C.; Thajchayapong, S.; Combining unsupervised anomaly detection and neural networks for driver identification. *Journal of Advanced Transportation*, vol. 2017, Article ID 6057830, 13 pp., 2017

Vlahogianni, E.I.; Karlaftis M.G.; Golias J.C.; Short-term traffic forecasting: where we are and where we're going, *Transportation Research Part C*, vol. 43, pp. 3-19, 2014.

Yang, H.; Akiyama, T.; Sasaki, T.; A neural network approach to the identification of real time origin-destination flows from traffic counts, *Proceeding of Int. Conf. on Artificial Intelligence Applications in Transportation Engineering*, San Buenaventura, CA, 1992.

Yasdi, R.; Prediction of road traffic using a neural network approach, *Neural Computing & Applications*, vol. 8, pp. 135-142, 1999.

Zhang, H.J.; Ritchie, S.G.; Lo, Z.P.; Macroscopic modeling of freeway traffic using an artificial neural network, *Transportation Research Record*, vol. 1588, pp. 110–119, 1997.

Zheng, F.; Van Zuylen, H.; Urban link travel time estimation based on sparse probe vehicle data, *Transportation Research Part C*, vol. 31, pp. 145-157, 2013.

Zheng Zhu, J.; Xin Cao, J.; Zhu, Y.; Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. *Transportation Research Part C*, vol. 47, pp. 139-154, 2014.

Zheng, F.; Van Zuylen, H.; Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C*, vol. 31, pp. 145-157, 2013.

Zheng, W.; Lee, D.H.; Shi, Q.; Short-term freeway traffic prediction: bayesian combined neural network approach, *Journal of Transportation Engineering*, vol. 132, pp. 114-121, 2006.

3 USO DELLE RETI NEURALI ARTIFICIALI PER L'ESTENSIONE SPAZIALE DEI FLUSSI DI TRAFFICO SU RETE STRADALE URBANA

3.1 Introduzione

La conoscenza in tempo reale dei flussi di traffico è essenziale per l'attuazione di qualsiasi strategia di controllo sulle reti stradali, sia in ambito urbano che extraurbano. I sistemi di trasporto intelligenti (ITS) non possono fare a meno dei sistemi di monitoraggio del traffico, che devono rilevare i dati in tempo reale e trasmetterli a una sala di controllo e a un sistema informatico centralizzato per consentire l'attuazione di strategie di controllo.

Un sistema di monitoraggio del traffico stradale si basa su sensori (o detector), opportunamente posizionati sulla rete, che misurano alcune caratteristiche del traffico, principalmente il flusso stradale (veh/h), la densità (veh/km) e la velocità (km/h).

Questi sensori, soprattutto in contesti urbani dove le reti sono significativamente congestionate, non possono essere presenti su tutti gli archi della rete, ma solo su alcuni di essi; di solito, maggiore è il numero di archi monitorati, maggiore è l'accuratezza della previsione del traffico e, quindi, l'efficacia delle strategie di controllo. Il costo dipende in modo significativo dal numero di archi monitorati; anche se i costi dei sensori di traffico sono diminuiti nell'ultimo decennio, il compromesso tra il numero di sensori e l'affidabilità e la precisione del sistema di monitoraggio è un fattore importante da considerare.

In questo capitolo, viene illustrata una metodologia per estendere spazialmente i dati di traffico ottenuti sugli archi monitorati ad altri archi della rete, in modo da aumentare la qualità dell'intero sistema limitandone il costo. Più in dettaglio, per raggiungere questo risultato sono state utilizzate le Reti Neurali Artificiali (ANN) che, una volta addestrate, stimano i flussi di traffico sugli archi non monitorati a partire dai flussi sugli archi monitorati. In precedenza, questo approccio è stato studiato solo in modo preliminare e su una rete di piccole dimensioni Gallo et al. (2016).

Il capitolo è articolato come segue: nel paragrafo 3.2 esaminiamo le tecnologie per i sistemi di monitoraggio, nel paragrafo 3.3 viene descritto l'approccio proposto, che viene testato su una rete in scala reale nel paragrafo 3.4. Infine, nel paragrafo 3.5 verranno riassunte le conclusioni.

3.2 Sistemi e tecnologie di monitoraggio del traffico stradale

In letteratura, i sistemi e le tecnologie di monitoraggio del traffico stradale sono stati ampiamente studiati da diversi punti di vista. Alcuni libri e articoli si sono concentrati sugli aspetti tecnologici dei sensori, mentre altri si sono occupati maggiormente delle reti di sensori e delle loro applicazioni agli ITS.

I manuali della FHWA (2006) hanno esaminato molti tipi di sensori di traffico, esplorandone la tecnologia, le applicazioni, le tecniche di installazione, la manutenzione, ecc. Bennet et al. (2006) hanno riassunto le tecnologie di raccolta dati per la gestione delle strade, compresi i sensori di traffico. Anche la nota di Leduc (2008) ha riassunto i metodi di raccolta dei dati sul traffico stradale. Salvo et al. (2014) e Li et al. (2018) hanno studiato l'uso degli UAV (*Unmanned Aerial Vehicles*) per il monitoraggio del traffico.

Un'interessante indagine sulle reti di sensori è stata riportata da Chong et al. (2003) che, tra l'altro, hanno identificato il controllo del traffico come uno dei principali campi in cui utilizzare con profitto queste tecnologie e, in particolare, le reti di sensori wireless. Hanno anche scritto: *"Tuttavia, questi sensori e la rete di comunicazione che li collega sono costosi; quindi, il monitoraggio del traffico è generalmente limitato a pochi punti critici"*. Nonostante i costi siano recentemente diminuiti, ancora oggi sono monitorati solo alcuni archi "critici", nella migliore delle ipotesi.

Tubaishat et al. (2009) hanno analizzato le tecnologie di rete di sensori wireless esistenti per gli ITS, identificando le seguenti applicazioni principali: a) monitoraggio dei parcheggi; b) monitoraggio e controllo del traffico; c) stima del traffico. Per quanto riguarda la stima del traffico, hanno identificato due problemi principali da risolvere: c.1) stimare il traffico in luoghi non monitorati; c.2) prevedere il traffico futuro. Per risolvere il problema c.1), Guitton et al. (2007) hanno proposto un metodo basato sull'apprendimento, utilizzando il coefficiente di correlazione di Pearson. Altri due metodi sono stati proposti da Tubaishat et al. (2009): un metodo statistico basato su modelli gaussiani e un metodo basato sulla stima della matrice origine-destinazione. Il problema c.2), invece, è stato più ampiamente studiato in letteratura; per questo problema sono state ampiamente adottate anche le ANN. Un'altra rassegna di reti di sensori wireless applicate agli ITS è proposta da Sharma et al. (2011). Alcuni interessanti aspetti tecnologici per la gestione in tempo reale dei centri dati che possono essere utilizzati negli ITS si trovano nei lavori di Shojafar et al. (2015), Cordeschi et al. (2015), e Baccarelli et al. (2016).

Alcuni lavori si sono concentrati su sensori "mobili"; Sun et al. (2004) hanno studiato l'applicazione di sensori ottici su veicoli stradali, mentre D'Acierno et al. (2009) hanno proposto di utilizzare gli autobus urbani come veicoli sonda per stimare le condizioni del traffico urbano. I dati dei veicoli sonda sono stati utilizzati da Klunder et al. (2017) per migliorare le prestazioni della rete con consigli di instradamento individuali in auto e da Yang et al. (2017) per stimare l'origine e la destinazione degli spostamenti.

3.3 Approccio proposto

In questa ricerca si propone l'utilizzo delle ANN per estendere spazialmente i dati dei flussi di traffico. Si ipotizza che alcuni archi di una rete stradale siano monitorati con appositi sensori e che altri archi, spesso una grande percentuale, non siano monitorati. L'obiettivo della ANN è prevedere i flussi di traffico sugli archi non monitorati a partire dai dati raccolti sugli archi monitorati. Lo stesso risultato può essere ottenuto utilizzando la stima della matrice origine-destinazione dai conteggi di traffico Bell (1983) e Cascetta (1984) come segue: 1) raccolta dei dati sui flussi di traffico sugli archi monitorati; 2) correzione della matrice OD utilizzando i dati raccolti; 3) assegnazione della matrice OD al modello di rete per stimare i flussi di traffico su tutti i collegamenti della rete. Questo approccio richiede tempi di calcolo elevati a causa delle procedure di correzione della matrice OD e di assegnazione del traffico; con una ANN addestrata, invece, i tempi di calcolo sono molto brevi, come si vedrà di seguito. Inoltre, esiste un ampio dibattito sulla (non) affidabilità della procedura di correzione della matrice OD (Marzano et al., 2009). I punti cruciali del metodo sono l'addestramento della ANN e la verifica del funzionamento dell'approccio in termini di previsione dei flussi di traffico.

È stato utilizzato un approccio di apprendimento supervisionato, in cui gli insiemi di dati di esempio sono generati tramite simulazione con la seguente procedura (vedi Figura 3.1):

- costruzione di un modello di rete stradale del caso di studio, il più dettagliato possibile;
- individuazione in questa rete di archi monitorati, e di archi non monitorati;
- generazione in modo casuale di diverse matrici origine-destinazione (OD), d^i con $i = 1, \dots, n$;
- assegnazione di ogni matrice, d^i , al modello di rete stradale, in modo da generare i corrispondenti flussi di traffico sui collegamenti monitorati, f_{ml}^i , e sui collegamenti non monitorati, f_{ul}^i con $i = 1, \dots, n$; si assume che questi siano i

flussi reali sulla rete e che costituiscano i dataset per l'addestramento, la validazione e il test della ANN; in particolare, i dataset di input contengono f_{ml}^i e i dataset di output contengono f_{ul}^i ;

- si dividono i dataset in tre gruppi: set di addestramento ($i = 1, \dots, t$), set di validazione ($i = t+1, \dots, t+v$) e set di test ($i = t+v+1, \dots, t+v+p$; $t+v+p=n$); i primi due gruppi sono utilizzati nella procedura di addestramento, mentre il terzo gruppo è utilizzato per valutare la capacità di previsione delle ANN addestrate.

Inoltre, due aspetti critici da considerare sono la progettazione della ANN e il metodo casuale per generare le matrici OD. In teoria, possono essere proposte infinite configurazioni di ANN per risolvere lo stesso problema. Esistono diversi tipi di ANN, come le reti *feedforward* singole o multistrato e le reti ricorrenti. Concentrandoci sulle reti *feedforward*, esse possono essere strutturate considerando un diverso numero di strati e di neuroni per ogni strato; quindi, in generale, è necessario testare diverse configurazioni per trovare la migliore per il problema specifico. A questo proposito, Philipp e Carbonell (2017) hanno proposto un metodo per determinare automaticamente la dimensione ottimale di una rete neurale.

Per quanto riguarda la generazione casuale delle matrici OD, si propongono due metodi. Il primo genera casualmente il valore di ogni cella della matrice OD tra zero e un valore massimo che può essere scelto in funzione delle informazioni sulla domanda di spostamento. Il secondo metodo genera casualmente il valore di ogni cella della matrice OD tra un valore minimo e un valore massimo che può essere diverso per ogni cella; si può adottare questo approccio solo se è disponibile una stima della matrice OD reale. Nella maggior parte dei casi, queste matrici stimate sono disponibili da analisi del traffico condotte nella stessa area di studio in cui deve essere implementato il sistema di monitoraggio. Infatti, poiché la domanda di viaggio segue alcuni modelli (ad esempio, la domanda tra alcune origini e destinazioni è alta mentre tra altre è bassa), tenerne in considerazione nell'approccio proposto dovrebbe migliorare i risultati. In questo caso, è necessario addestrare diverse ANN con riferimento a diversi periodi (ora di punta del mattino, ora di punta del pomeriggio, ora non di punta e così via). Di seguito, verranno testati entrambi i metodi per ottenere le matrici OD utilizzate per generare i dataset di esempio e quattro diverse ANN a singolo strato, che differiscono per il numero di neuroni.

3.4 Sperimentazione e risultati

Sono state effettuate due sperimentazioni, una sull'intera rete di Benevento e l'altra sui singoli quartieri della stessa rete.

3.4.1 Test su intera rete stradale della città di Benevento

Il caso di studio si basa sulla rete stradale di Benevento, una città di circa 60.000 abitanti nel sud Italia. Il modello di rete stradale (vedi Figura 3.2) è molto dettagliato ed è composto da 949 segmenti stradali (1.577 archi orientati), 678 nodi e 80 centroidi (matrice con 6400 coppie OD); i centroidi rappresentano nel modello i punti di origine e destinazione degli spostamenti tra le zone. Questo modello di rete rappresenta 216 km di strade. Tutti gli elementi sono stati modellati tenendo conto delle loro caratteristiche, come la lunghezza, la velocità e la capacità degli archi e i metodi di regolazione alle intersezioni (semaforo, stop, precedenza, rotatoria, ecc.). Inoltre, sono disponibili tre matrici OD nei seguenti periodi: ora di punta mattutina (MP), ora di punta pomeridiana (AP) e ora non di punta (OP); tali matrici sono state stimate utilizzando modelli matematici e rilievi del traffico stradale durante la redazione del Piano Urbano del Traffico di Benevento. In particolare, i rilievi di traffico sono stati effettuati nelle 139 sezioni stradali riportate in Figura 3.2; nei nostri test, assumiamo che il sistema di monitoraggio del traffico preveda i sensori in queste sezioni stradali: gli archi corrispondenti rappresentano, quindi, i collegamenti monitorati. Gli altri collegamenti reali della rete, invece, saranno considerati i collegamenti non monitorati. I dati sui flussi di traffico stradale sono stati pre-elaborati eliminando dall'insieme degli archi non monitorati quelli con flussi di traffico nulli; è necessario rimuovere questi archi poiché il modello di rete stradale a volte non è in grado di stimare i bassi flussi di traffico sulle strade locali e potrebbe falsare i risultati. Nel nostro problema, quindi, consideriamo 139 archi monitorati e 1.186 archi non monitorati.

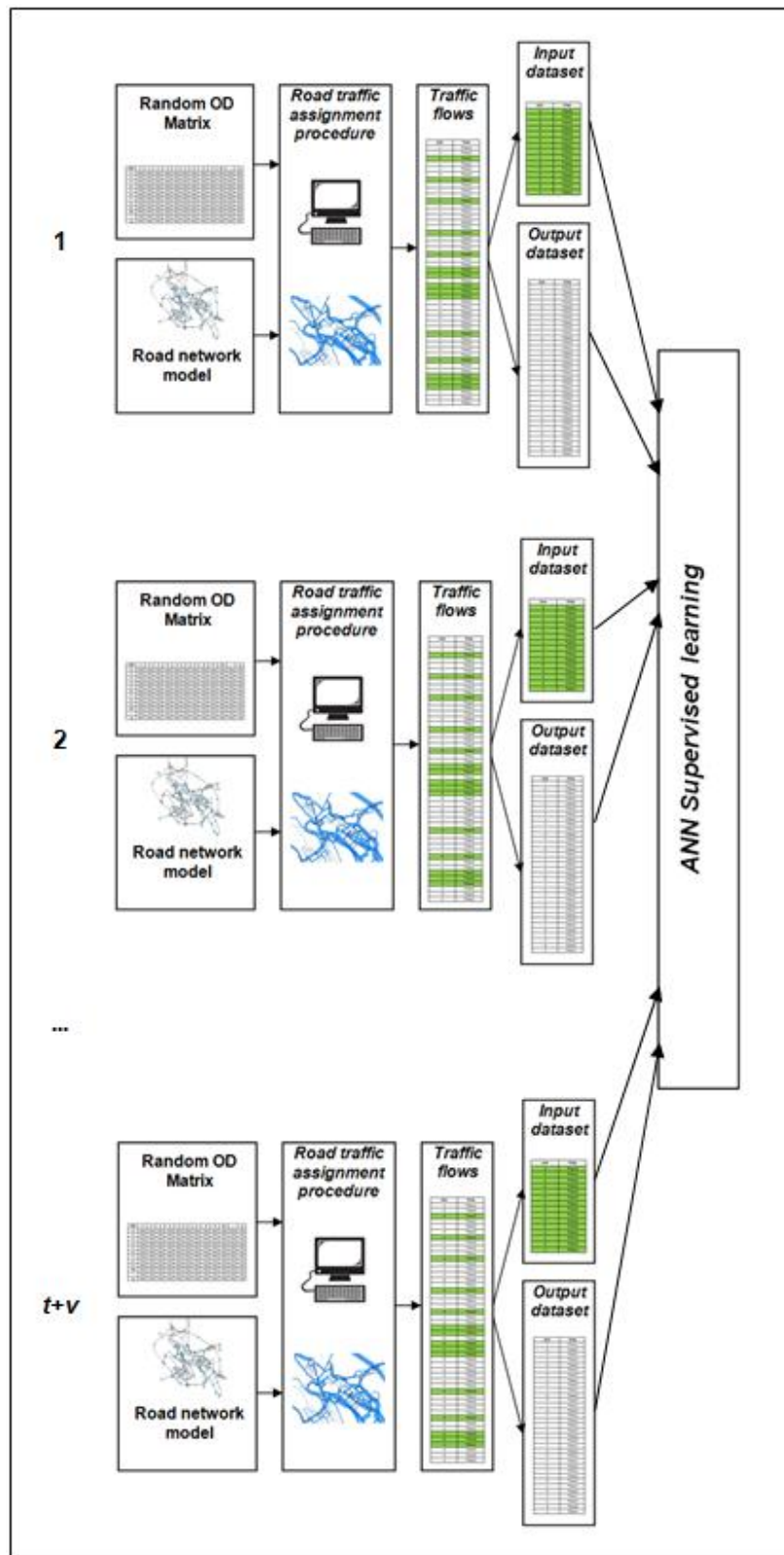


Figura 3.1 - Generazione dei set di dati di addestramento

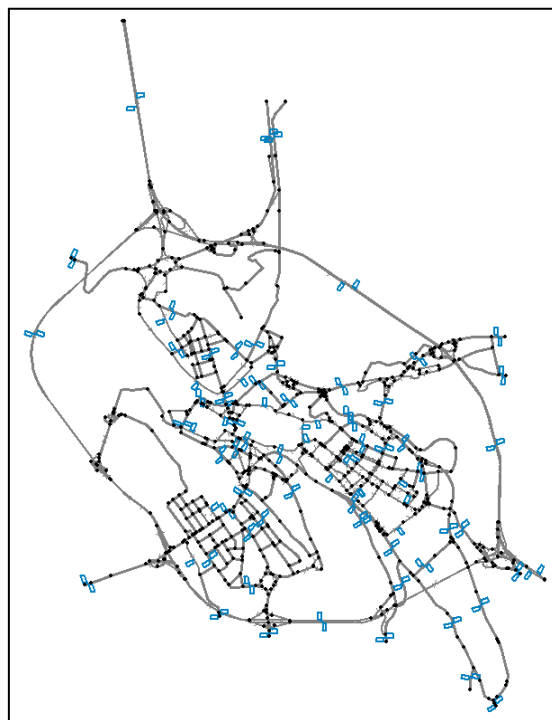


Figura 3.2 - Modello di rete stradale e archi monitorati

Per il caso di studio, abbiamo testato diverse ANN, addestrate con diversi set di dati. Il primo set di dati di addestramento è stato ottenuto partendo da matrici OD generate in modo casuale, scegliendo un valore massimo del flusso di OD, senza considerare i modelli della domanda di viaggio (lo stesso valore massimo per tutte le coppie di OD). Questo valore massimo è diverso nei diversi periodi (MP, AP e OP). È necessario utilizzare questo approccio quando non sono disponibili le stime delle matrici OD. Il secondo dataset di addestramento, invece, è stato generato perturbando il flusso di domanda su ogni coppia OD (in base alle matrici OD disponibili), moltiplicando il suo valore stimato per un numero casuale compreso tra 0,8 e 1,2; questo approccio è possibile solo se si dispone della stima della matrice OD di ogni periodo (MP, AP e OP). In questo modo, la struttura della domanda è presa in considerazione in modo esplicito.

Complessivamente, sono state generate 180 matrici, con il primo metodo (senza considerare la domanda stimata), e 600 matrici, 200 per ogni periodo (MP, AP e OP), con il secondo metodo (considerando la domanda stimata); ogni matrice è stata assegnata alla rete stradale in modo da generare un dataset di flussi di traffico. Nel primo caso, 150 dataset di flussi di traffico sono stati utilizzati per l'addestramento della ANN e gli altri per verificare la qualità dei risultati. Nel secondo caso, 190 dataset di flussi di traffico sono stati utilizzati per l'addestramento della RNA nel periodo corrispondente e gli altri

per la verifica. Infine, nel secondo caso sono stati utilizzati 570 dataset di flussi di traffico (190 per ogni periodo) per addestrare una ANN, considerando i tre periodi contemporaneamente. Sono state addestrate ANN con diverse strutture in ciascun caso; tutte le ANN sono feedforward con un singolo strato, ma con un numero diverso di neuroni: 6, 10, 20 e 50. La Tabella 3.1 riassume le 20 ANN addestrate.

Demand Pattern	Period	Neurons
No	All	6
		10
		20
		50
Yes	MP	6
		10
		20
		50
	AP	6
		10
		20
		50
	OP	6
		10
		20
		50
	All	6
		10
		20
		50

Tabella 3.1 – RNA addestrate

Per valutare la capacità delle ANN di stimare i flussi di traffico stradale sugli archi non monitorati sono state utilizzate tre misure di errore di previsione: errore quadratico medio (MSE), radice dell'errore quadratico medio (RMSE) e errore quadratico medio relativo (RMSE%). L'RMSE% permette di confrontare i risultati corrispondenti a diversi livelli di domanda e, quindi, è utilizzato per identificare i casi migliori e peggiori. Nei nostri test, queste misure sono calcolate come segue:

$$MSE = (1/n_{ul}) \cdot \sum_{j=1, \dots, n_{ul}} (f_j^{sim} - f_j^{ANN})^2$$

$$RMSE = (MSE)^{1/2}$$

$$RMSE\% = RMSE / (\sum_{j=1, \dots, n_{ul}} f_j^{ANN} / n_{ul})$$

dove n_{ul} è il numero di collegamenti non monitorati, f_j^{sim} è il flusso di traffico sul collegamento j ottenuto con la simulazione (il flusso reale, secondo le nostre ipotesi) e f_j^{ANN} è il flusso di traffico stimato con la ANN addestrata.

Inoltre, abbiamo calcolato il coefficiente di determinazione, R^2 , come segue:

$$R^2 = 1 - (\sum_{j=1, \dots, n} (f_j^{sim} - f_j^{ANN})^2 / \sum_{j=1, \dots, n} (f_j^{sim} - M(f^{sim}))^2)$$

dove $M(f^{sim})$ è la media dei flussi di traffico simulati. Il coefficiente di determinazione è utilizzato per valutare la capacità di un modello di replicare i dati osservati; il valore massimo di R^2 è 1 (capacità perfetta del modello di riprodurre i dati osservati: $f_j^{sim} = f_j^{ANN}$ ogni link j) e quanto più R^2 si avvicina a 1, tanto maggiore è la bontà del modello calibrato. Anche se nel nostro caso non abbiamo un modello esplicito, possiamo utilizzare il coefficiente di determinazione per misurare la capacità della ANN di riprodurre i flussi. Naturalmente, tutte queste misure sono state calcolate con dataset non utilizzati nella fase di addestramento. Nella Tabella 3.2. riassumiamo i risultati principali, riportando per ogni ANN i casi migliori e peggiori. Nella Tabella 3.3 sono riportate le epoche e i tempi di calcolo per l'addestramento delle ANN; si può notare che la procedura di addestramento è veloce, richiedendo da alcuni secondi fino a 1 minuto e 37 secondi. L'utilizzo delle ANN addestrate per la stima dei flussi di traffico richiede, invece, meno di un secondo in tutti i casi.

Nella Figura 3.3 sono riportati i diagrammi di dispersione dei casi peggiori e migliori delle ANN addestrate senza considerare il modello di domanda. Al contrario, la Figura 3.4 riporta gli stessi diagrammi per le ANN addestrate considerando il modello di domanda. Tutte le figure si riferiscono alle ANN con sei neuroni, che sono quelle che hanno dato risultati migliori in termini di RMSE% e R^2 .

Neurons		6		10		20		50	
		Best Case	Worst Case	Best Case	Worst Case	Best Case	Worst Case	Best Case	Worst Case
MSE									
Without demand pattern	All periods	4068.57	2940.26	5750.85	8618.68	8398.84	4395.26	10,902.83	3817.52
With demand pattern	MP	2745.55	12,187.61	2711.22	13,885.76	2794.76	12,428.60	3125.40	13,403.28
	AP	1837.69	6275.93	1788.36	8857.49	2253.55	8686.23	2723.28	8990.26
	OP	352.64	1577.61	319.70	1606.36	312.06	1436.70	487.15	2332.71
	All periods	3273.54	2638.57	2487.71	1855.52	3073.77	2360.71	2842.60	1961.30
R^2									
Without demand pattern	All periods	0.966	0.786	0.946	0.668	0.931	0.702	0.912	0.762
With demand pattern	MP	0.978	0.908	0.978	0.895	0.977	0.897	0.973	0.890
	AP	0.975	0.925	0.976	0.895	0.971	0.893	0.963	0.906
	OP	0.975	0.894	0.976	0.889	0.976	0.902	0.966	0.846
	All periods	0.975	0.831	0.979	0.877	0.977	0.847	0.978	0.871
RMSE									
Without demand pattern	All periods	63.79	54.22	92.84	75.83	91.65	66.30	104.42	61.79
With demand pattern	MP	52.40	110.40	52.07	117.84	52.87	111.48	55.91	115.77
	AP	42.87	79.22	42.29	94.11	47.47	93.20	52.19	94.82
	OP	18.78	39.72	17.88	40.08	17.67	37.90	22.07	48.30
	All periods	57.21	51.37	49.88	43.08	55.44	48.59	53.32	44.29
RMSE%									
Without demand pattern	All periods	0.00016	0.00034	0.00022	0.00044	0.00023	0.00040	0.00027	0.00038
With demand pattern	MP	0.00013	0.00027	0.00013	0.00029	0.00013	0.00028	0.00014	0.00029
	AP	0.00014	0.00025	0.00014	0.00030	0.00015	0.00030	0.00017	0.00031
	OP	0.00013	0.00027	0.00013	0.00028	0.00013	0.00027	0.00016	0.00034
	All periods	0.00014	0.00033	0.00013	0.00029	0.00013	0.00032	0.00013	0.00030

Tabella 3.2 – Misure dell'errore di precisione

Neurons		6		10		20		50	
		Epochs	Time	Epochs	Time	Epochs	Time	Epochs	Time
MSE									
Without demand pattern	All periods	447	16''	465	17''	544	22''	489	26''
With demand pattern	MP	406	17''	196	8''	322	14''	387	21''
	AP	185	8''	146	6''	201	9''	216	16''
	OP	197	8''	300	13''	583	25''	596	30''
	All periods	1000	1'11''	1000	1'13''	1000	1'16''	1000	1'37''

Tabella 3.3 – Tempi di calcolo

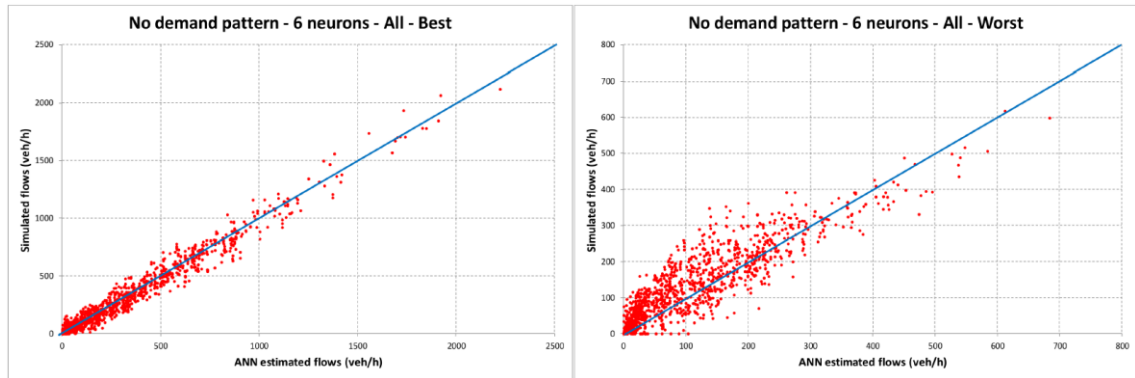


Figura 3.3 - Diagrammi di dispersione con matrici casuali per la ANN con sei neuroni

I risultati riportati nella Tabella 3.2 e nella Figura 3.3 evidenziano un'ottima capacità delle RNA di stimare i flussi di traffico sugli archi non monitorati, se si considerano i modelli di domanda. Infatti, i valori di RMSE% sono molto bassi e i valori di R^2 sono vicini a 1 anche nei casi peggiori. Senza considerare i modelli di domanda (si veda la Tabella 3.2 e la Figura 3.3), i risultati possono comunque essere considerati accettabili per questo tipo di problema, anche se gli errori sono più significativi. Infatti, i modelli di simulazione del traffico sono affetti da alcuni errori inevitabili dovuti a i) le caratteristiche degli archi stradali (tempi di percorrenza, velocità, capacità, ecc.) non possono essere modellate con elevata precisione e alcune di esse (tempo di percorrenza e velocità) dipendono da flussi con funzioni che devono essere correttamente calibrate; ii) i flussi di traffico stradale, che influenzano anche le prestazioni degli archi, sono sempre affetti da variazioni stocastiche; iii) le origini e le destinazioni reali degli spostamenti sono sparse sul territorio (i punti possibili sono teoricamente infiniti) mentre il modello deve concentrarle solo in alcuni punti specifici (centroidi). Per questi motivi, di solito, i modelli di ingegneria dei trasporti con $R^2 \geq 0,75$ sono considerati del tutto accettabili. Inoltre, si può notare che le ANN con sei neuroni hanno quasi sempre le prestazioni migliori.

Questi risultati portano ad alcune indicazioni pratiche per l'implementazione dell'approccio proposto in casi reali: a) considerare esplicitamente le matrici OD di partenza è necessario per ottenere un'ottima accuratezza ma, se le stime delle matrici non sono disponibili, i risultati possono essere considerati comunque accettabili; b) le ANN con pochi neuroni (6 o 10) hanno mostrato le migliori prestazioni; pertanto, non è necessario costruire e addestrare RNA con molti neuroni.

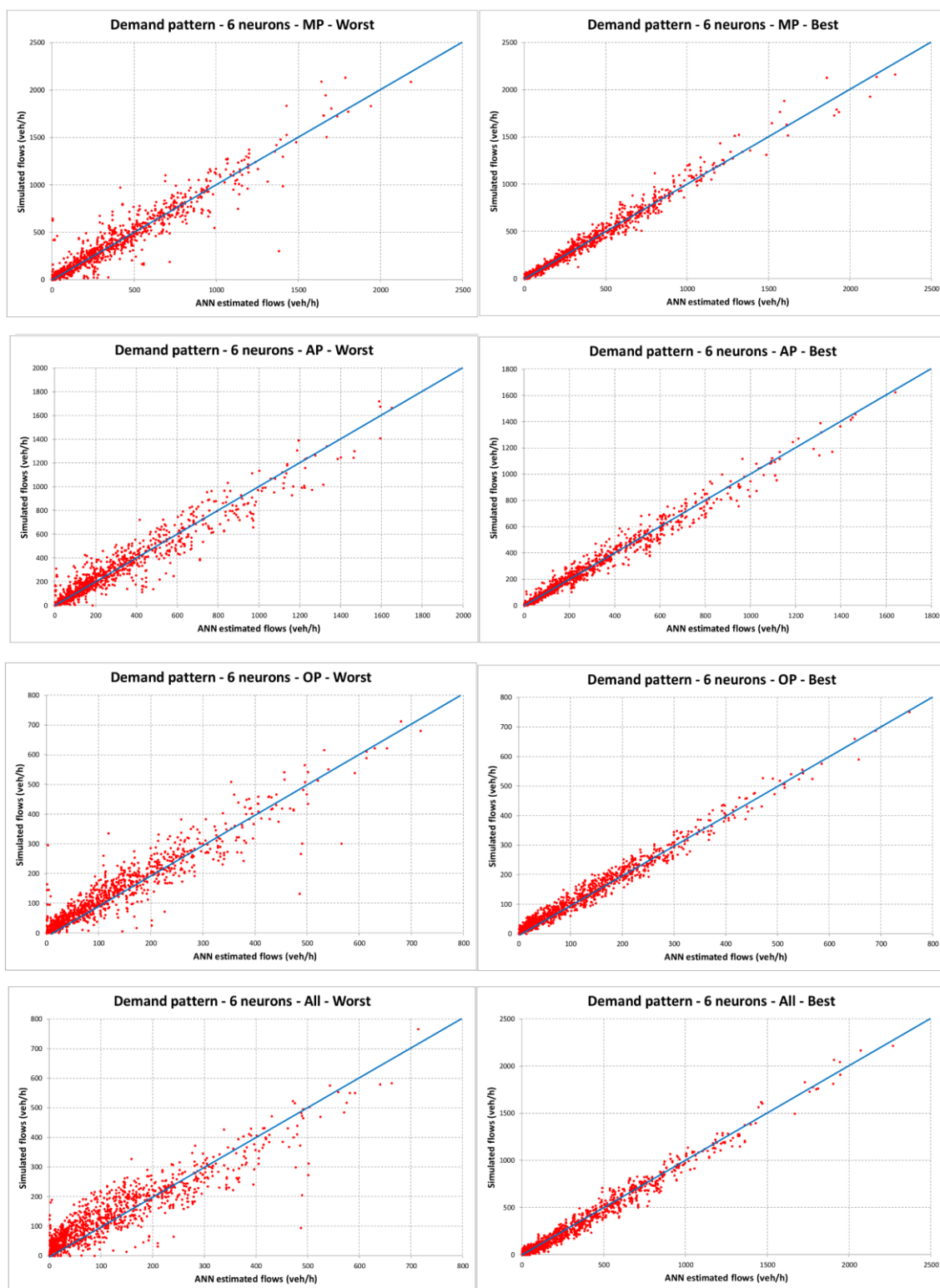


Figura 3.4 - Diagrammi di dispersione con matrici perturbate per le RNA con sei neuroni (MP: ora di punta del mattino; AP: ora di punta del pomeriggio; OP: ora non di punta; All: tutti i periodi)

3.4.2 Test su singoli quartieri della città di Benevento

In questo caso, la rete stradale è stata suddivisa in 6 zone (vedi Figura 3.5), che corrispondono ai quartieri della città.

Una ANN feedforward è stata addestrata per ogni zona e ogni periodo, generando 36 casi. In accordo con i risultati ottenuti sull'intera rete, è stata testata una ANN con uno strato nascosto e sei neuroni. La fase di addestramento è stata eseguita con 180 dataset di esempio, mentre altri 10 dataset sono stati conservati per verificare la bontà dei risultati. La fase di addestramento ha richiesto tempi di calcolo da 3 secondi (Zona Pacevecchia_AP) a 3 minuti (Zona Centro_OP), con un PC i7-7700HQ, 2,80 GHz, RAM 16 GB. Nella Tabella 3.5 sono riassunti i risultati ottenuti, riportando: coefficienti di determinazione (R^2) per i 10 dataset utilizzati per la verifica (minimo, medio, massimo e deviazione standard).

Nelle Figure 3.6-3.11 sono riportati i diagrammi di dispersione relativi ai casi migliori e peggiori per ciascuna zona. Confrontando i risultati con quelli relativi all'intera città, si può notare che la capacità previsionale della ANN aumenta limitando l'area oggetto di studio (cfr. Figura 3.12). Questi risultati sono interessanti in vista di future applicazioni a casi reali: è utile identificare piccole zone in cui i dati raccolti dai rilevatori di traffico stradale possono essere utilizzati per prevedere i flussi su altri collegamenti appartenenti alla stessa zona.

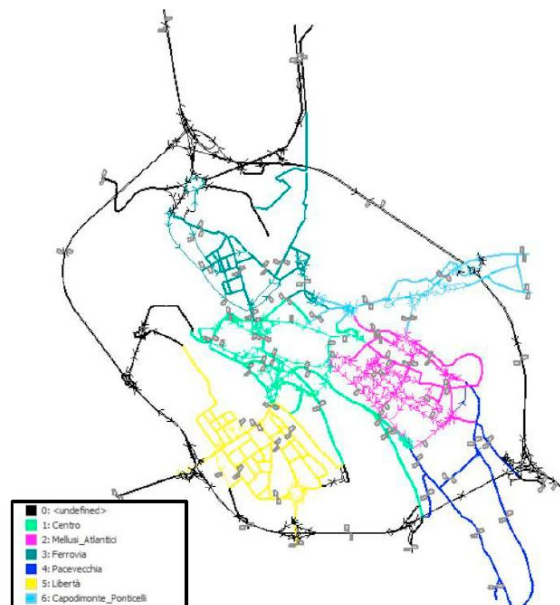


Figura 3.5 – Rete stradale e sua suddivisione in zone

Zone	Monitored links	Unmonitored links
Centro	28	190
Mellusi_Atlantici	22	179
Ferrovia	19	150
Pacevecchia	18	65
Libertà	16	250
Capodimonte_Ponticelli	10	122

Tabella 3.4 – Caratteristiche delle zone

		Coefficient of determination (R^2)			
Period	Zone	Max	Min	Average	Standard deviation
MP	Centro	0.997	0.989	0.994	0.002
	Mellusi_Atlantici	0.992	0.982	0.989	0.003
	Ferrovia	0.996	0.986	0.991	0.004
	Pacevecchia	0.997	0.985	0.991	0.004
	Libertà	0.995	0.985	0.991	0.003
	Capodimonte_Ponticelli	0.991	0.972	0.983	0.007
AP	Centro	0.997	0.992	0.995	0.001
	Mellusi_Atlantici	0.990	0.974	0.981	0.005
	Ferrovia	0.998	0.983	0.993	0.004
	Pacevecchia	0.986	0.973	0.981	0.004
	Libertà	0.996	0.989	0.993	0.002
	Capodimonte_Ponticelli	0.992	0.976	0.988	0.005
OP	Centro	0.996	0.992	0.994	0.001
	Mellusi_Atlantici	0.990	0.979	0.986	0.004
	Ferrovia	0.995	0.987	0.991	0.003
	Pacevecchia	0.993	0.974	0.984	0.006
	Libertà	0.996	0.986	0.993	0.003
	Capodimonte_Ponticelli	0.994	0.976	0.987	0.006

Tabella 3.5 – Risultati principali

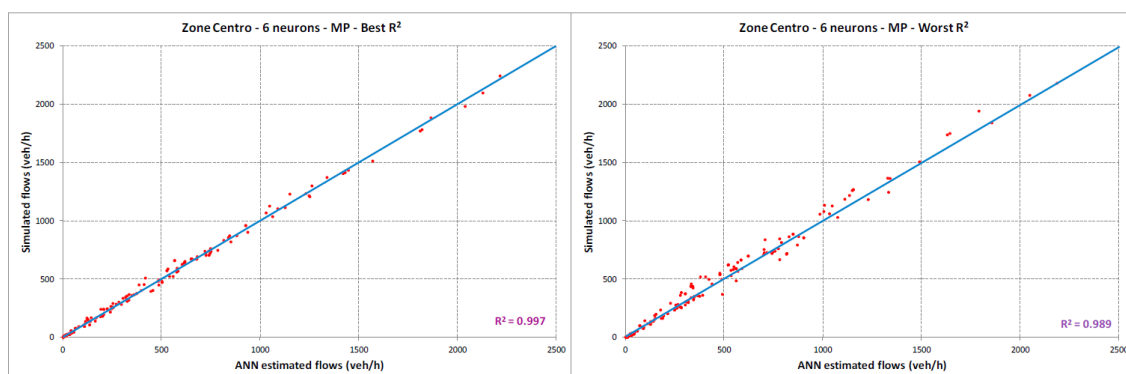


Figura 3.6 – Casi migliori e peggiori per la zona Centro

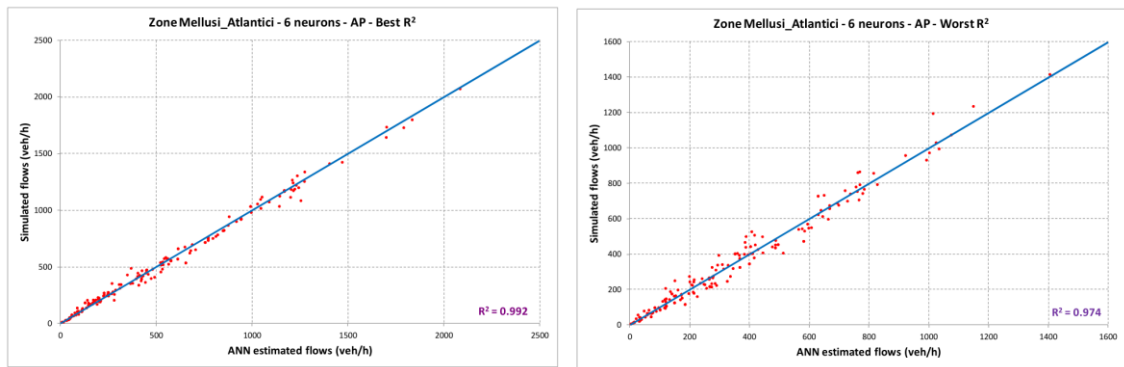


Figura 3.7 - Casi migliori e peggiori per la zona Mellusi/ Atlantici

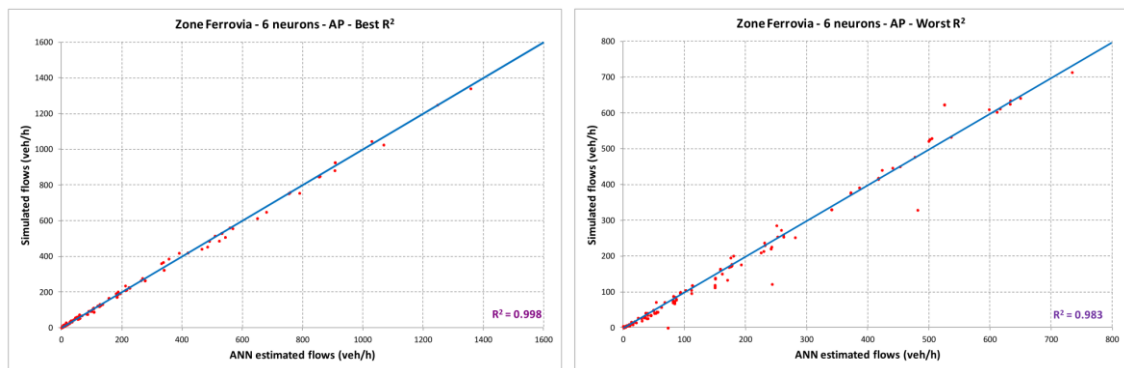


Figura 3.8 - Casi migliori e peggiori per la zona Ferrovia

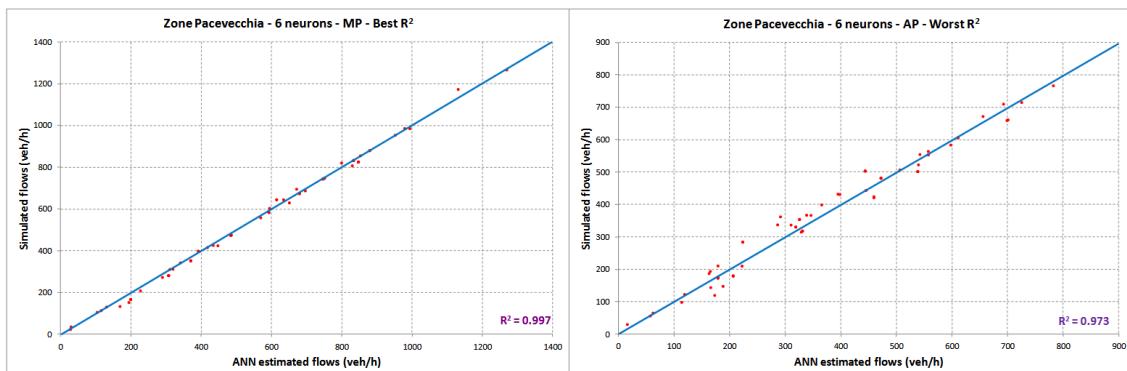


Figura 3.9 - Casi migliori e peggiori per la zona Pacevecchia

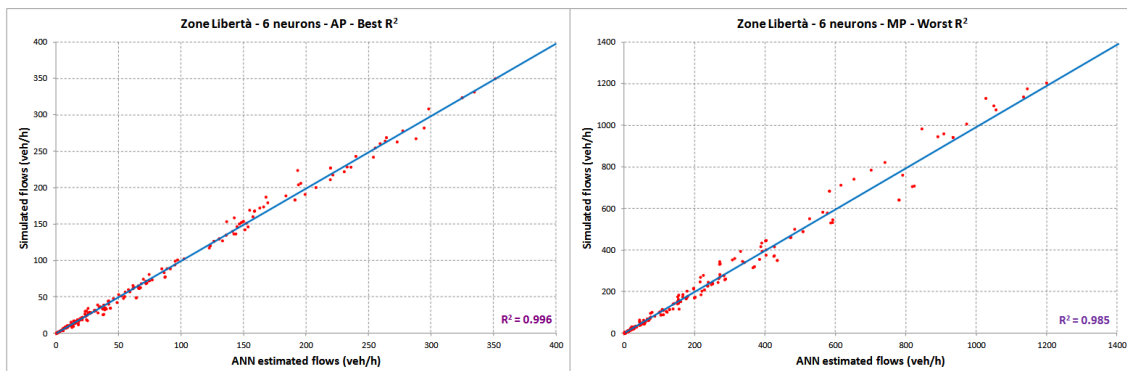


Figura 3.10 - Casi migliori e peggiori per la zona Libertà

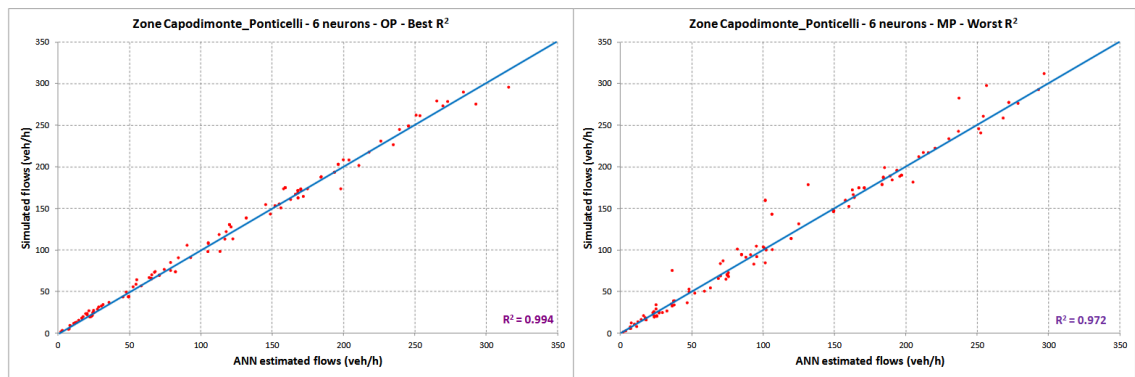


Figura 3.11 - Casi migliori e peggiori per la zona Ponticelli

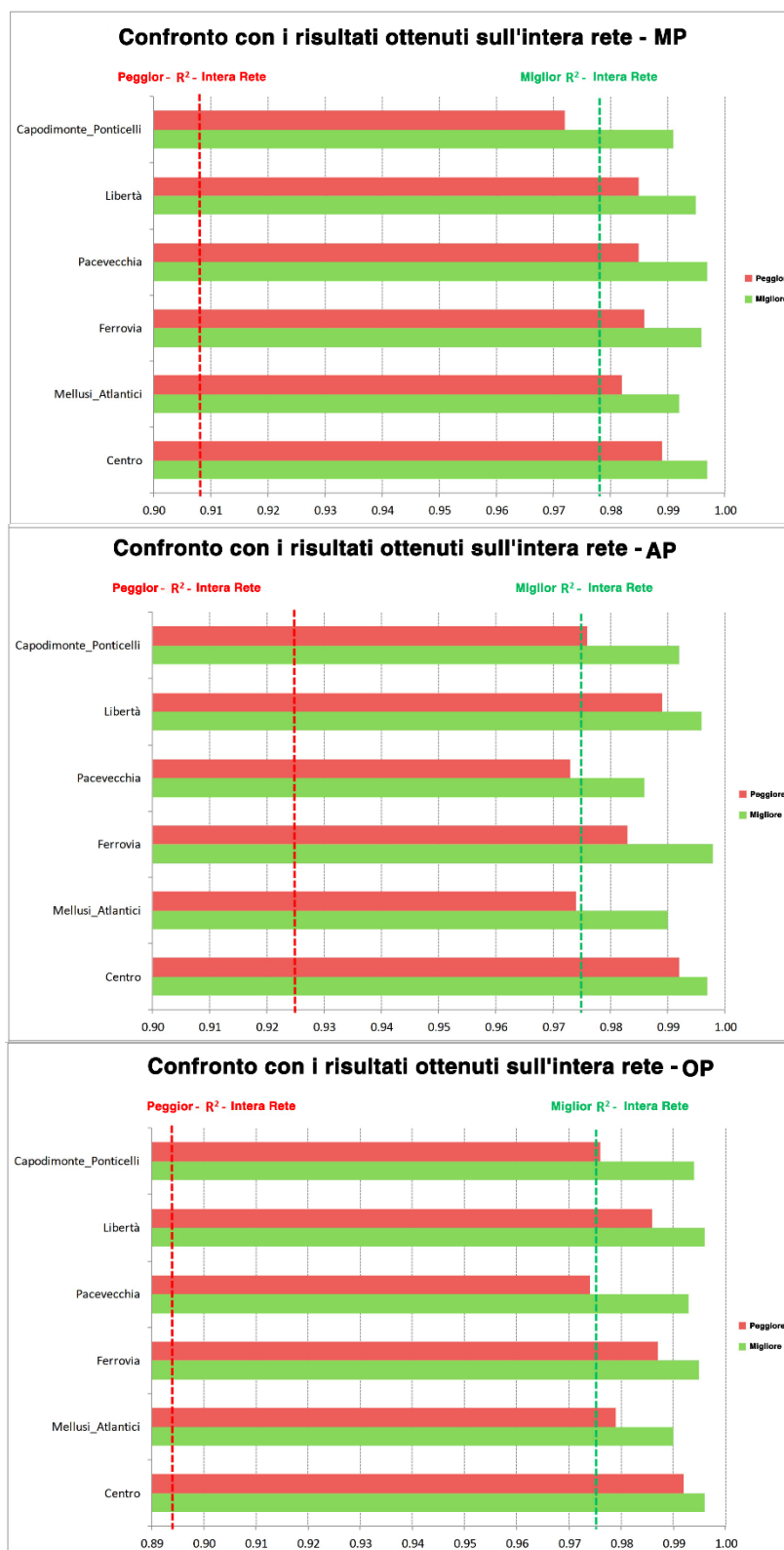


Figura 3.12 - Confronto con i risultati ottenuti sull'intera rete

3.5 Conclusioni

In questo capitolo è stata studiata l'estensione spaziale dei flussi di traffico stradale da alcuni archi monitorati ad altri archi non monitorati utilizzando un approccio basato sulle Reti Neurali Artificiali. A differenza della previsione a breve termine, l'estensione spaziale è stata studiata in letteratura solo marginalmente. L'estensione spaziale può consentire di ridurre il numero di sensori in un sistema di monitoraggio, controllando lo stesso numero di archi.

Per la generazione dei dataset di addestramento viene proposto un metodo basato sulla simulazione del traffico e l'approccio metodologico è stato testato su una rete in scala reale. Vengono adottati due metodi diversi per la generazione delle matrici OD necessarie a generare i dataset: considerare o meno matrici OD disponibili; nel primo caso, l'approccio proposto ha restituito ottimi risultati, mentre nel secondo caso i risultati possono essere considerati, comunque, accettabili.

Sono state utilizzate le ANN per la previsione dei flussi, considerando due casi studio:

- applicazione all'intera rete della città di Benevento, utilizzando come dati di input e output quelli relativi all'intera rete cittadina;
- applicazione ai quartieri in cui la città è stata suddivisa, utilizzando come dati di input e output quelli relativi solo ad una parte della rete.

I risultati riassunti nella Tabella 3.5 e nella Figura 3.12 sottolineano i vantaggi della riduzione del numero di dati e della limitazione del territorio in analisi. Infatti, i coefficienti di determinazione calcolati per ogni quartiere sono più alti rispetto al caso dell'intera città e le differenze tra i loro valori minimi e massimi sono più basse, mostrando una buona stabilità e fruibilità dei risultati.

Il buon funzionamento del metodo non deve trarre in inganno; infatti, l'addestramento delle reti neurali è ottenuto utilizzando set di dati simulati. I dati reali, invece, sono affetti da variazioni stocastiche che potrebbero ridurre la capacità predittiva delle reti neurali. Tuttavia, si ritiene che i risultati ottenuti siano molto promettenti, perché i valori dei coefficienti di determinazione sono molto alti, e si può pensare che le perturbazioni stocastiche ridurranno la precisione del metodo, permettendo comunque una buona capacità predittiva.

La ricerca futura potrà essere indirizzata a testare altre reti su scala reale e ANN multistrato, a testare diverse funzioni di trasferimento, a verificare i miglioramenti che si possono ottenere nelle procedure di correzione della matrice OD aggiungendo i flussi di

traffico stimati a quelli rilevati, a ottimizzare la posizione dei sensori e a identificare gli archi non monitorati che sono meglio correlati a quelli monitorati, considerando solo questi come output del problema. Potrà, inoltre, essere rivolta a testare il metodo di addestramento delle ANN con dati reali, anziché simulati, a proporre metodi di deep-learning e a combinare l'estensione spaziale con la previsione a breve termine.

Bibliografia Capitolo 3

- Baccarelli, E.; Cordeschi, N.; Mei, A.; Panella, M.; Shojafar, M.; Stefa, J.; Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study. *IEEE Network*, vol. 30, pp. 54-61, 2016.
- Bell, M.G.H.; The estimation of origin–destination matrix from traffic counts. *Transportation Science*, vol. 10, pp. 198-217, 1983.
- Bennet, C.R.; de Solminihac, S.; Chamorro, A.; Data Collection Technologies for Road Management. *Transport Notes No. 30*, The World Bank, pp. 1-8, 2006.
- Cascetta, E.; Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. *Transportation Research Part B*, vol. 18, pp. 289-299, 1984.
- Chong, C.-Y.; Kumar, S.P.; Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, vol. 91, pp. 1247-1256, 2003.
- Cordeschi, N.; Amendola, D.; Baccarelli, E.; Reliable adaptive resource management for cognitive cloud vehicular networks. *IEEE Transactions on Vehicular Technology*, 2015, vol. 64, pp. 2528-2537.
- D'Acierno, L.; Carteni, A.; Montella, B.; Estimation of urban traffic conditions using an Automatic Vehicle Location (AVL) System. *European Journal of Operational Research*, vol. 196, pp. 719-736, 2009.
- FHWA; Traffic Detector Handbook: Third Edition - Volume I, U.S. Department of Transportation, Report No. FHWA-HRT-06-108, 2006.
- FHWA; Traffic Detector Handbook: Third Edition - Volume II, U.S. Department of Transportation, Report No. FHWA-HRT-06-139, 2006.
- Gallo, M.; Simonelli, F.; De Luca, G.; Della Porta C.; An Artificial Neural Network approach for spatially extending road traffic monitoring measures. *Proceeding of 2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS 2016)*, pp. 209-213, 2016.

Guitton, A.; Skordylis, A.; Trigoni, N.; Utilizing Correlations to Compress Time-Series in Traffic Monitoring Sensor Networks. *Proceedings of Wireless Communications and Networking Conference*, 2007.

Klunder, G.A.; Taale, H.; Kester, L.; Hoogendoorn, S.; Improvement of network performance by in-vehicle routing using floating car data. *Journal of Advanced Transportation*, vol. 2017, Article ID 8483750, pp. 16, 2017.

Leduc, G.; Road Traffic Data: Collection Methods and Applications. *Working Papers on Energy, Transport and Climate Change*, JRC 47967, 2008.

Li, M.; Zhen, L.; Wang, S.; Lv, W.; Qu, X.; Unmanned aerial vehicle scheduling problem for traffic monitoring. *Computers & Industrial Engineering*, vol. 122, pp. 15-23, 2018.

Marzano, V.; Papola, A.; Simonelli, F.; Limits and perspectives of effective O–D matrix correction using traffic counts. *Transportation Research Part C*, vol. 17, pp. 120-132, 2009.

Philipp, G.; Carbonell, J.G.; Nonparametric neural networks. *5th International Conference on Learning Representations*, Toulon, France, 2017.

Salvo, G.; Caruso, L.; Scordo, A.; Urban Traffic Analysis through an UAV. *Procedia - Social and Behavioral Sciences*, vol. 111, pp. 1083-1091, 2014.

Sharma, A.; Chaki, R.; Bhattacharya, U.; Applications of wireless sensor network in Intelligent Traffic System: A review. *Proceeding of 3rd International Conference on Electronics Computer Technology*, pp. 53-57, 2011.

Shojafar, M.; Cordeschi, N.; Amendola, D.; Baccarelli, E.; Energy-saving adaptive computing and traffic engineering for real-time-service data centers. *Proceedings of 2015 IEEE International Conference on Communication Workshop (ICCW)*, 8-12 June 2015, pp. 1800-1806, 2015.

Shojafar, N.; Amendola, D.; Baccarelli, E.; Reliable adaptive resource management for cognitive cloud vehicular networks. *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 2528-2537, 2015.

Sun, Z.; Bebis, G.; Miller, R.; On-road vehicle detection using optical sensors: a review. *Proceedings of the 7th IEEE Conference on Intelligent Transportation Systems*, pp. 585-590, 2004.

Tubaishat, M.; Zhuang, P.; Qi, Q.; Shang, Y.; Wireless sensor networks in intelligent transportation systems. *Wireless Communications and Mobile Computing*, vol. 9, pp. 287-302, 2009.

Yang, X.; Lu, Y.; Hao, W.; Origin-destination estimation using probe vehicle trajectory and link counts. *Journal of Advanced Transportation*, vol. 2017, Article ID 4341532, pp. 18, 2017.

4 RETI NEURALI ARTIFICIALI PER LA PREVISIONE DEI FLUSSI DI PASSEGGERI SU LINEE DI METROPOLITANA

4.1 Introduzione e articolazione del capitolo

In questo capitolo si propone l'uso delle Reti Neurali Artificiali (ANN) per la previsione dei flussi di passeggeri a bordo delle metropolitane in funzione del conteggio dei passeggeri ai tornelli delle stazioni. Si ipotizza che i tornelli delle stazioni della metropolitana registrino il numero di passeggeri in ingresso mediante un sistema di conteggio automatico e che questi dati siano disponibili in intervalli limitati (ad esempio, aggregazione temporale di 15 minuti); l'obiettivo è stimare i passeggeri a bordo su ogni tratta della linea (cioè, tra due stazioni successive) in funzione dei dati dei tornelli raccolti nei periodi precedenti. La scelta dell'intervallo temporale varia con gli orari di servizio, l'accesso è controllato da tornelli in grado di contare i passeggeri che entrano in ogni stazione identificando o meno la loro direzione (nella maggior parte delle stazioni esistenti, ad eccezione dei terminali, i passeggeri possono salire sui treni in direzioni diverse). Si propone un metodo basato su reti neurali artificiali (ANN) per stimare il numero di passeggeri su ogni segmento di una linea metropolitana utilizzando i dati registrati ai tornelli. Infatti, i dati a bordo sono raramente disponibili in tempo reale perché richiedono scanner a infrarossi o, in alternativa, sensori basati sul peso, sulle immagini o sulle fotocellule che raramente sono installati sulle carrozze.

Le reti neurali artificiali sono addestrate utilizzando dati di simulazione ottenuti con una procedura di carico dinamico della linea ferroviaria. L'approccio proposto è stato testato su un caso reale: la linea 1 della metropolitana di Napoli. I risultati numerici mostrano che l'approccio proposto è in grado di prevedere i flussi sulle tratte della metropolitana con una precisione soddisfacente.

Il capitolo è articolato come segue: il paragrafo 4.2 riporta lo stato dell'arte; il paragrafo 4.3 descrive il problema da risolvere e l'approccio ANN; il metodo adottato per generare i dati utilizzati nei test numerici è riportato nel paragrafo 4.4; il paragrafo 4.5 descrive il caso di studio e i risultati; il paragrafo 4.6 conclude e identifica alcune prospettive di ricerca.

4.2 Stato dell'arte

Il contributo della letteratura alla previsione del flusso di passeggeri delle metropolitane è interessante. I metodi di deep learning sono stati proposti in Liu e Chen (2017) e testati su un sistema Bus Rapid Transit (BRT). Il modello proposto prevede il flusso orario, adottando un'architettura di deep learning a tre stadi. Questo articolo individua quattro diversi approcci: (1) algoritmi classici tradizionali; (2) modelli regressivi; (3) modelli basati sull'apprendimento automatico, comprese le ANN; (4) modelli ibridi. Tutti i casi studiati, tuttavia, si riferiscono a periodi di tempo a breve o lungo termine, senza considerare l'estensione spaziale. Tra questi, i casi riportati da Tsai et al. (2009) e Li e Yang (2007) sono applicati alle ferrovie e si basano su ANN, concentrandosi rispettivamente sulla previsione a breve e a lungo termine. La previsione a breve termine nelle metropolitane urbane è stata studiata anche con altri metodi, come il filtro di Kalman (Jiao et al., 2016) e i modelli ARIMA (AutoRegressive Integrated Moving Average) (Cai et al., 2014).

Li et al. (2017) hanno proposto una funzione a base radiale multiscala (MSRBF) per la previsione dei flussi di passeggeri delle metropolitane a breve termine in occasioni speciali, come eventi sportivi, concerti, ecc. In questo caso, il flusso di passeggeri è molto irregolare e le previsioni sono più difficili da ottenere. Ling, et al. (2018) hanno utilizzato i dati delle smart card per prevedere i flussi di passeggeri nella metropolitana di Shenzhen (Cina), analizzando quattro modelli predittivi: un modello di media storica, un modello ANN, un modello di regressione e un modello ad albero di regressione con gradient-boost. Liu et al. (2019) hanno proposto un metodo di deep learning per la previsione a breve termine dei flussi di passeggeri in entrata e in uscita dalla metropolitana, mentre Wang et al. (2019) hanno proposto un modello Novel Markov-Grey per risolvere lo stesso problema.

4.3 Descrizione del problema e approccio ANN

Si ipotizza che i tornelli controllino tutti gli accessi ad una linea metropolitana: ogni utente, entrando in una stazione, utilizza un biglietto (o un pass) per attraversare il tornello. Inoltre, i tornelli sono in grado di contare solo gli utenti che entrano nella stazione senza collegare l'origine di ogni viaggio con la destinazione corrispondente. Questa situazione è comune a molte linee metropolitane, come la Linea 1 della

metropolitana di Napoli (Italia), che sarà oggetto del test su scala reale. Infatti, i tornelli sono spesso installati solo per facilitare il controllo/validazione dei biglietti ed evitare i viaggi senza biglietto e, in contesti urbani, la tariffa è la stessa indipendentemente dalla coppia origine-destinazione. Di seguito, consideriamo due casi: (a) tornelli all'ingresso della stazione che misurano solo i passeggeri in entrata, senza alcuna indicazione sulla direzione da seguire; (b) tornelli all'accesso ai binari che forniscono anche informazioni sulla direzione del viaggio (vedi Figura 4.1).

I dati raccolti ai tornelli possono essere utilizzati, con un basso investimento tecnologico, per implementare un sistema di monitoraggio dell'intera linea metropolitana, generando informazioni sui passeggeri di ogni tratta ferroviaria (tra due stazioni). Tali informazioni possono essere di grande utilità per gli operatori della metropolitana per l'attuazione di strategie in tempo reale, come l'aumento o la riduzione della frequenza, la determinazione della composizione dei treni (numero di carrozze passeggeri), la programmazione di corse aggiuntive e così via.

Il problema da risolvere è la stima dei carichi sulla linea a partire dai dati dei tornelli. A questo scopo, si propongono delle ANN feedforward, che sono adatte perché (a) non è necessario conoscere esplicitamente le relazioni tra ingressi e uscite, (b) i risultati si ottengono rapidamente, e (c) le condizioni al contorno di solito non cambiano tanto da invalidare le previsioni.

La struttura delle ANN prevede uno strato di input con un nodo per ogni tornello, uno strato di output con un nodo per ogni carico del convoglio e uno o più strati nascosti. La struttura delle ANN va identificata per ogni singolo caso. Un punto cruciale riguarda la natura dinamica del problema: il treno si muove lungo la linea, caricando e scaricando passeggeri nelle stazioni in diversi intervalli di tempo. Pertanto, il numero di passeggeri a bordo tra due stazioni al tempo t dipende dai passeggeri caricati e scaricati nelle stazioni precedenti in tempi diversi. Pertanto, gli input della ANN devono considerare i conteggi dei tornelli riferiti a diversi intervalli di tempo precedenti a quelli oggetto di previsione. Il numero di input dipenderà dalla durata del viaggio tra i capolinea. L'altro punto cruciale è la fase di addestramento delle ANN; qui proponiamo un metodo di apprendimento supervisionato, in cui gli insiemi di dati di esempio sono generati attraverso modelli di simulazione dinamici. Si può notare che non è possibile utilizzare dati reali per la fase di addestramento. Infatti, solo i dati di ingresso (sui passeggeri che entrano nelle stazioni) sono disponibili, mentre i dati di uscita (passeggeri a bordo) possono essere conosciuti

solo se tutte le carrozze hanno sensori in grado di misurarli; in questo caso, però, l'approccio proposto non sarebbe utile.

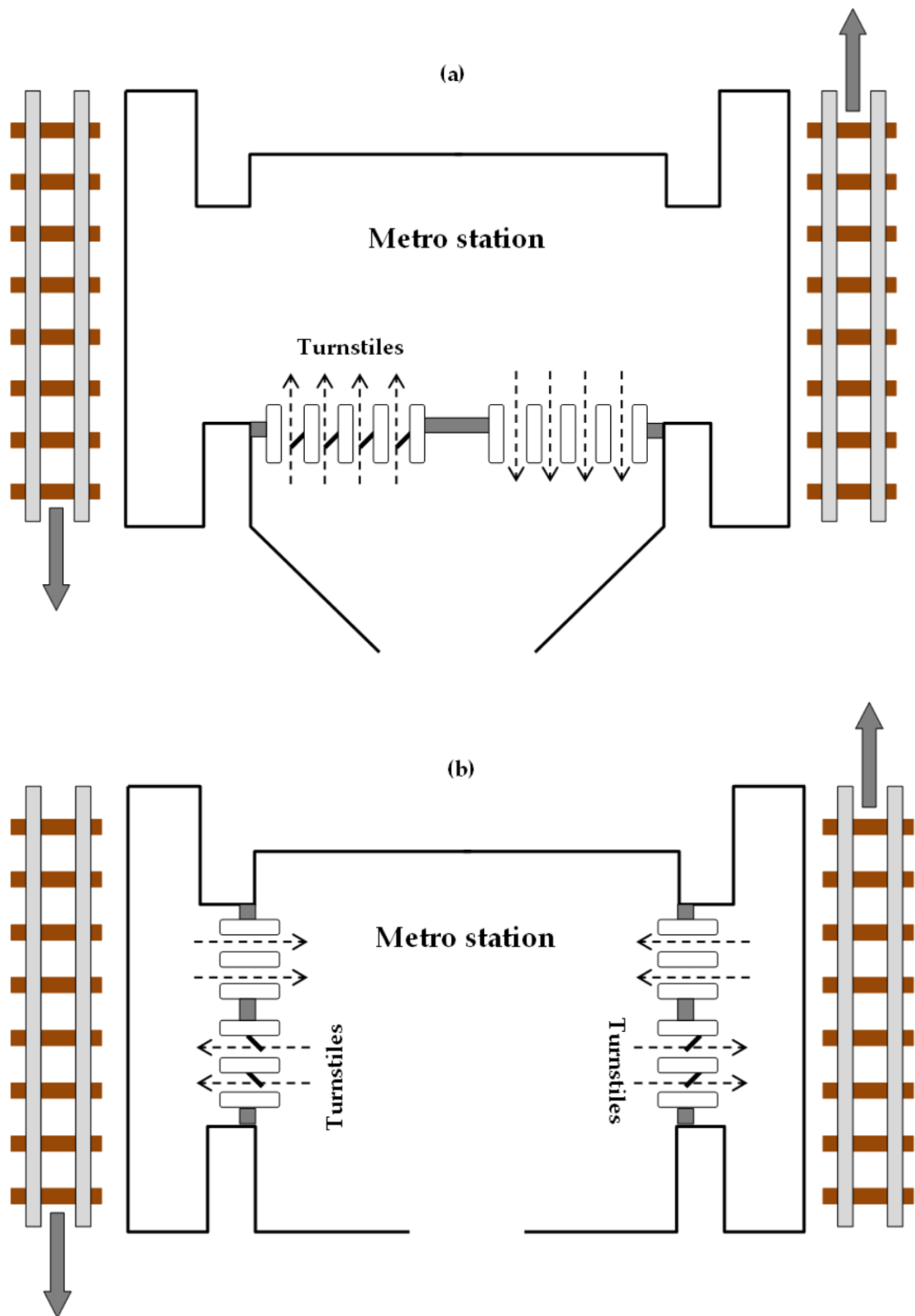


Figura 4.1 - Tipi di stazioni della metropolitana: (a) tornelli all'ingresso; (b) tornelli agli accessi alle banchine.

4.4 Generazione dei set di dati di addestramento

Per generare i set di dati di addestramento, si è scelto il modello di simulazione proposto da D’Acerno et al. (2018). Questo modello presuppone che:

1. le banchine possano ospitare tutti i passeggeri (in arrivo, in attesa e in partenza);
2. in ogni stazione e per ogni direzione sia presente un solo binario;
3. il tempo di sosta sia costante e indipendente dal numero di passeggeri che scendono e salgono;
4. non ci sia interazione sulla banchina tra i passeggeri che scendono, salgono e aspettano;
5. la capacità di ciascun treno sia fissa;
6. i passeggeri siano distribuiti uniformemente tra le carrozze del treno;
7. non ci sia interazione nel treno tra i passeggeri che scendono, salgono e sono a bordo.

In questo caso di studio, assumiamo che i passeggeri seguano una regola FIFO (First In-First Out) per salire a bordo del convoglio. I dettagli analitici del modello sono disponibili in D’Acerno et al. (2018).

Dataset →	1	2	...	ds
<i>Input data</i>				
Turnstile 1–Period $t - 1$	$c^1_{1,t-1}$	$c^2_{1,t-1}$	...	$c^{ds}_{1,t-1}$
Turnstile 2–Period $t - 1$	$c^1_{2,t-1}$	$c^2_{2,t-1}$	...	$c^{ds}_{2,t-1}$
...	...	...	...	...
Turnstile ts –Period $t - 1$	$c^1_{ts,t-1}$	$c^2_{ts,t-1}$	...	$c^{ds}_{ts,t-1}$
Turnstile 1–Period $t - 2$	$c^1_{1,t-2}$	$c^2_{1,t-2}$	...	$c^{ds}_{1,t-2}$
Turnstile 2–Period $t - 2$	$c^1_{2,t-2}$	$c^2_{2,t-2}$	...	$c^{ds}_{2,t-2}$
...	...	...	...	...
Turnstile ts –Period $t - 2$	$c^1_{ts,t-2}$	$c^2_{ts,t-2}$	...	$c^{ds}_{ts,t-2}$
Turnstile 1–Period $t - 3$	$c^1_{1,t-3}$	$c^2_{1,t-3}$	...	$c^{ds}_{1,t-3}$
Turnstile 2–Period $t - 3$	$c^1_{2,t-3}$	$c^2_{2,t-3}$	...	$c^{ds}_{2,t-3}$
...	...	...	...	...
Turnstile ts –Period $t - 3$	$c^1_{ts,t-3}$	$c^2_{ts,t-3}$	...	$c^{ds}_{ts,t-3}$
Turnstile 1–Period $t - 4$	$c^1_{1,t-4}$	$c^2_{1,t-4}$	...	$c^{ds}_{1,t-4}$
Turnstile 2–Period $t - 4$	$c^1_{2,t-4}$	$c^2_{2,t-4}$	...	$c^{ds}_{2,t-4}$
...	...	...	...	...
Turnstile ts –Period $t - 4$	$c^1_{ts,t-4}$	$c^2_{ts,t-4}$	...	$c^{ds}_{ts,t-4}$
<i>Output data</i>				
Railway section 1–Period t	$f^1_{1,t}$	$f^2_{1,t}$	...	$f^{ds}_{1,t}$
Railway section 2–Period t	$f^1_{2,t}$	$f^2_{2,t}$	...	$f^{ds}_{2,t}$
...	...	...	...	...
Railway section rs –Period t	$f^1_{rs,t}$	$f^2_{rs,t}$	...	$f^{ds}_{rs,t}$

Tabella 4.1 - Struttura dei dataset di addestramento per il periodo t

Utilizzando questo modello, generiamo i dataset di addestramento per il caso studio come segue: (a) numerose matrici origine-destinazione (OD) riferite a intervalli di 15' sono state generate casualmente a partire da una matrice OD di base; (b) quattro matrici OD, riferite a quattro intervalli di tempo consecutivi, sono assegnate alla linea della metropolitana, ottenendo come risultati i passeggeri contati ai tornelli, per ogni intervallo, e i passeggeri a bordo in ogni tratta ferroviaria nell'intervallo di tempo successivo; (c) i dati di output del problema sono i passeggeri sulle tratte ferroviarie, mentre i dati di input sono i passeggeri contati ai tornelli in quattro intervalli di tempo, corrispondenti ai quattro periodi precedenti di 15' (ad esempio, i dati di input sono i passeggeri contati ai tornelli in quattro intervalli di tempo): i flussi sulle tratte ferroviarie tra le 10:15 e le 10:30 sono stimati in base ai conteggi ai tornelli negli intervalli: 9:15-9:30, 9:30-9:45, 9:45-10:00 e 10:00-10:15.

Pertanto, la struttura dei dataset di addestramento è riportata nella Tabella 4.1, dove si utilizzano le seguenti notazioni:

d_s è il numero di insiemi di dati;

t è il periodo di riferimento;

$c_{ij,t}$ è il numero di passeggeri al tornello j nel periodo t per il dataset i ;

$f_{ik,t}$ è il carico sulla tratta ferroviaria k nel periodo t per il dataset i .

4.5 Caso di studio e risultati numerici

L'approccio proposto è stato testato sulla Linea 1 della metropolitana di Napoli. Questa linea (vedi Figura 4.2), lunga 18 km e con 18 stazioni, collega i quartieri ad alta densità di Napoli ed è un'infrastruttura cruciale per la mobilità urbana.

Considerando queste caratteristiche, abbiamo 34 o 18 tornelli, a seconda se dividiamo i passeggeri in base alla direzione o meno, e 34 sezioni ferroviarie monodirezionali. Le caratteristiche principali della linea sono riassunte nella Tabella 4.2.

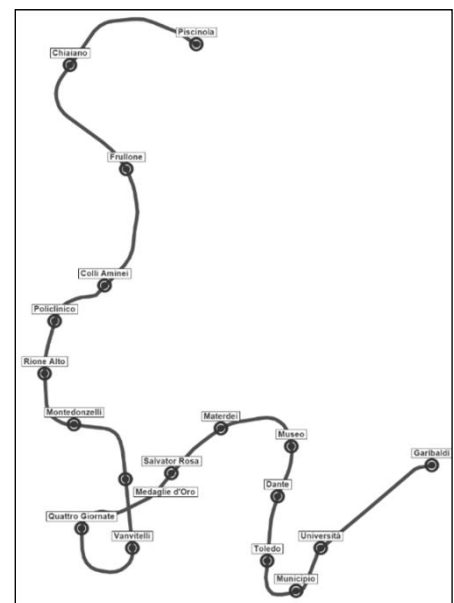


Figura 4.2 – percorso linea 1

Stazioni	18
Corse giorno feriale	241
Capacità del convoglio (pax/convoglio)	864
Lunghezza della linea (km) (direzione di andata/ritorno)	18.8/18.6
Distanza (min)	8-20

Tabella 4.2 – caratteristiche della linea 1

I dataset di addestramento sono stati ottenuti simulando 2500 matrici OD generate in modo casuale. Eliminandone alcune perché i risultati non erano affidabili (troppi passeggeri rispetto alla capacità effettiva), abbiamo generato 2279 dataset di addestramento. Di questi dataset, 2229 sono stati utilizzati per l'intero processo di addestramento delle ANN con il software MatLab: 1561 dataset di addestramento (70%), 334 dataset di validazione (15%) e 334 dataset di test (15%). I restanti 50 dataset sono stati utilizzati per verificare la bontà delle ANN addestrate con esempi non utilizzati in precedenza nel processo di addestramento. Sono state testate sei strutture di ANN per entrambi i casi: (a) tornelli all'ingresso della stazione (18 tornelli) e (b) tornelli ai punti di accesso ai binari (34 tornelli). In totale sono state testate e addestrate 12 ANN, come riportato nella Tabella 4. 3.

Caso	ANN	Nodi di ingresso	Nodi di uscita	Strati nascosti	Neuroni
a	a_1_6	72	34	1	6
a	a_1_10	72	34	1	10
a	a_1_20	72	34	1	20
a	a_2_6	72	34	2	6/6
a	a_2_10	72	34	2	10/10
a	a_2_20	72	34	2	20/20
b	b_1_6	136	34	1	6
b	b_1_10	136	34	1	10
b	b_1_20	136	34	1	20
b	b_2_6	136	34	2	6/6
b	b_2_10	136	34	2	10/10
b	b_2_20	136	34	2	20/20

Tabella 4.3 – Strutture della RNA

La fase di addestramento ha richiesto tempi di calcolo da 30 s (caso a_1_6) a 8 min (caso b_2_20), con un Personal Computer Hewlett Packard i7-7700HQ, 280 GHz, RAM 16 GB. Nella Tabella 4.4 sono riportati i migliori e i peggiori coefficienti di determinazione (R^2) per ogni caso, riferiti ai 50 dataset non utilizzati nella fase di addestramento, e le corrispondenti medie e varianze. I set di dati per i quali R^2 è inferiore a 0,9, 0,8, 0,7 e 0,6

sono riportati nella Tabella 4.5. In queste tabelle, i valori migliori per ciascuna ANN sono sottolineati.

Esaminando i risultati riportati nelle Tabelle 4.4 e 4.5, possiamo identificare come migliore struttura di ANN quella con uno strato nascosto e 20 neuroni per il caso (a), e due strati nascosti e 10 neuroni per il caso (b). I corrispondenti diagrammi di dispersione nei casi di migliore e peggiore R^2 sono riportati nelle Figure 4.3 e 4.4.

ANN	Migliore	Peggior	Media	Varianza
a_1_6	<u>0.9946</u>	0.5487	0.7984	0.0124
a_1_10	0.9941	0.4990	0.8108	0.0160
a_1_20	0.9931	<u>0.5613</u>	<u>0.8332</u>	0.0157
a_2_6	0.9916	0.5353	0.7949	<u>0.0121</u>
a_2_10	0.9930	0.4638	0.8136	0.0175
a_2_20	0.9933	0.5342	0.8244	0.0162
b_1_6	0.9875	0.5460	0.8016	<u>0.0129</u>
b_1_10	<u>0.9905</u>	0.3505	0.8115	0.0202
b_1_20	0.9882	0.4226	<u>0.8291</u>	0.0160
b_2_6	0.9785	<u>0.5119</u>	0.7775	0.0132
b_2_10	0.9889	0.4935	0.8221	0.0132
b_2_20	0.9852	0.4492	0.8075	0.0245

Tabella 4.4 – Coefficienti di determinazione (R^2)(i valori più alti sono sottolineati)

ANN	$R^2 < 0,9$	$R^2 < 0,8$	$R^2 < 0,7$	$R^2 < 0,6$
Numero di set di dati				
a_1_6	41	25	9	4
a_1_10	36	22	<u>7</u>	4
a_1_20	<u>33</u>	<u>18</u>	<u>7</u>	<u>3</u>
a_2_6	43	25	10	<u>3</u>
a_2_10	35	21	8	4
a_2_20	34	19	9	5
b_1_6	37	24	10	<u>2</u>
b_1_10	34	23	8	5
b_1_20	32	<u>19</u>	8	3
b_2_6	43	25	10	6
b_2_10	35	20	<u>7</u>	<u>2</u>
b_2_20	<u>30</u>	21	9	7
Percentuale di set di dati				
a_1_6	82%	50%	18%	8%
a_1_10	72%	44%	<u>14%</u>	8%
a_1_20	<u>66%</u>	<u>36%</u>	<u>14%</u>	<u>6%</u>
a_2_6	86%	50%	20%	<u>6%</u>
a_2_10	70%	42%	16%	8%
a_2_20	68%	38%	18%	10%
b_1_6	74%	48%	20%	<u>4%</u>
b_1_10	68%	46%	16%	10%
b_1_20	64%	<u>38%</u>	16%	6%
b_2_6	86%	50%	20%	12%
b_2_10	70%	40%	<u>14%</u>	<u>4%</u>
b_2_20	<u>60%</u>	42%	18%	14%

Tabella 4.5 – Analisi dei valori R^2

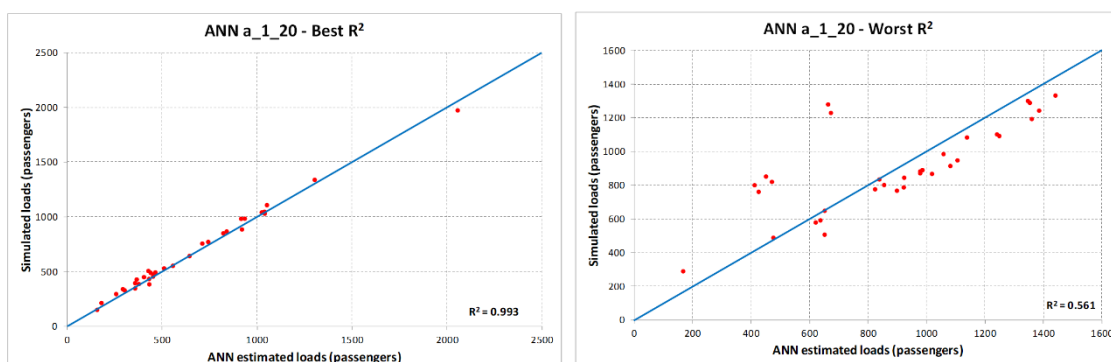


Figura 4.3 – Diagrammi di dispersione a_1_20 (migliore, diagramma superiore; peggiore, diagramma inferiore).

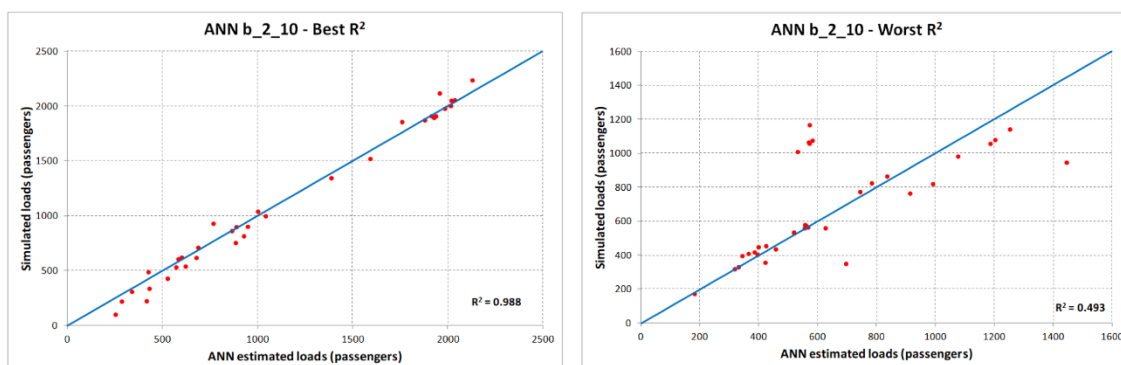


Figura 4.4 - Diagrammi di dispersione RNA b_2_10 (migliore, diagramma superiore; peggiore, diagramma inferiore)

4.6 Conclusioni

In questo lavoro si è affrontato lo studio della previsione dei flussi di passeggeri sulle tratte ferroviarie di una linea metropolitana a partire dai conteggi ai tornelli e si è proposto di utilizzare le reti neurali artificiali (ANN) per la sua soluzione. I dataset di addestramento sono stati generati utilizzando un modello di simulazione. Sono stati considerati due casi: tornelli all'ingresso delle stazioni e tornelli per l'accesso ai binari. Per entrambi sono state progettate e addestrate diverse ANN.

I risultati hanno mostrato una buona capacità delle ANN di prevedere i carichi sulle sezioni ferroviarie. Lo studio ci ha permesso di identificare la migliore struttura di ANN per ogni caso.

È importante sottolineare che il problema studiato è rilevante per la gestione in tempo reale delle linee metropolitane. I dati ai tornelli possono essere facilmente raccolti con metodi che non richiedono significativi investimenti aggiuntivi, mentre i dati ottenuti con la procedura sopra descritta (carichi sulle tratte di linea) sono essenziali per gli operatori

del servizio e, a differenza dei primi, non sono facilmente rilevabili in tempo reale e in modo continuo.

La ricerca futura potrebbe portare proficuamente in diverse direzioni. Innanzitutto, si potrebbero testare altre strutture di ANN. Poi, il problema potrebbe essere esteso a sistemi metropolitani più complessi, compresi quelli con più di una linea. Infine, si potrebbero studiare altri metodi e approcci di deep-learning da applicare a questo problema.

Bibliografia Capitolo 4

Cai, C.; Yao, E.; Wang, M.; Zhang, Y.; Prediction of urban railway station's entrance and exit passenger flow based on multiply ARIMA model. *J. Beijing Jiaotong Univ.*, vol. 38, pp. 135-140, 2014.

D'Acierno, L.; Botte, M.; Montella, B.; Assumptions and simulation of passenger behaviour on rail platforms. *Int. J. Transp. Dev. Integr.*, vol. 2, pp. 123-135, 2018.

Jiao, P.; Li, R.; Sun, T.; Hou, Z.; Ibrahim, A.; Three revised kalman filtering models for short-term rail transit passenger flow prediction. *Math. Probl. Eng.* vol. 2016, pp. 1-10, 2016.

Li, J.T.; Yang, J.F.; Prediction of Dalian station passenger based on RBF neural network. *J. Dalian Jiaotong Univ.*, vol. 28, pp. 32-34, 2007.

Li, Y.; Wang, X.; Sun, S.; Ma, X.; Lu, G.; Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Research Part C*, vol. 77, pp. 306-328, 2017.

Ling, X.; Huang, Z.; Wang, C.; Zhang, F.; Wang, P.; Predicting subway passenger flows under different traffic conditions. *PLoS ONE*, vol. 13, pp. 1-23, 2018.

Liu, L.; Chen, R.-C.; A novel passenger flow prediction model using deep learning methods. *Transp. Res. Part. C*, vol. 84, pp. 74-91, 2017.

Liu, Y.; Liu, Z.; Jia, R.; DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C*, vol. 101, pp. 18-34, 2019.

Tsai, T.H.; Lee, C.K.; Wei, C.H.; Neural network based temporal feature models for short-term railway passenger demand forecasting. *Expert Syst. Appl.*, vol. 36, pp. 3728-3736, 2009.

Wang, Y.; Ma, J.; Zhang, J.; Metro Passenger Flow Forecast with a Novel Markov-Grey Model. *Period. Polytech. Transp. Eng.*, 2019.

5 CONCLUSIONI

Questo lavoro ha proposto, sperimentato e valutato l'utilizzo delle Reti Neurali Artificiali (ANN) per l'estensione spaziale dei dati sui flussi di traffico e di utenti. Due applicazioni sono state implementate e sperimentate; una relativa ai flussi veicolari in una rete stradale urbana e l'altra relativa ai flussi di utenti su una linea di metropolitana. Entrambe le proposte sono state sperimentate su casi a scala reale.

L'importanza delle applicazioni proposte risiede nella possibilità di ottenere più dati ed informazioni sulla rete a parità di sensori di traffico o di punti di rilievo. Nel caso stradale, è possibile, dai dati rilevati su alcuni archi della rete, ricavare i dati su altri archi, senza la necessità di rilievi aggiuntivi. Nel caso della rete di metropolitana è, invece, possibile ottenere informazioni sui carichi dei convogli che percorrono la linea in ogni tratta in funzione dei flussi rilevati alle stazioni.

Entrambe le metodologie proposte possono essere molto utili nell'ambito dei sistemi di monitoraggio e controllo del traffico e dei flussi, consentendo, a parità di costo delle installazioni tecnologiche, di avere più informazioni.

Mentre l'utilizzo delle ANN è stato ampiamente trattato in letteratura per l'estensione temporale dei dati di traffico, in particolare per le previsioni a breve termine, l'utilizzo delle stesse per l'estensione spaziale è stato molto poco studiato e le applicazioni proposte sono entrambe originali.

I risultati ottenuti hanno mostrato un'ottima capacità delle ANN di riprodurre i flussi di traffico e di utenti sugli archi non monitorati della rete in funzione dei dati disponibili sugli archi monitorati. I valori degli indicatori di performance (R^2 , RMSE, ecc.) hanno confermato la validità della procedura e la sua applicabilità a casi reali.

Pur potendo considerare questo lavoro completo, per gli obiettivi prefissi, diverse sono le direzioni di ricerca che possono essere intraprese e approfondite.

La ricerca futura potrà essere indirizzata a testare altre reti su scala reale e ANN multistrato, a testare diverse funzioni di trasferimento, a verificare i miglioramenti che si possono ottenere nelle procedure di correzione della matrice OD aggiungendo i flussi di traffico stimati a quelli rilevati, a ottimizzare la posizione dei sensori e a identificare i collegamenti non monitorati che sono meglio correlati a quelli monitorati, considerando solo questi collegamenti come output del problema. La ricerca potrà essere inoltre rivolta a testare il metodo di addestramento delle ANN con dati reali anziché simulati, a proporre

metodi di deep-learning e a combinare l'estensione spaziale con la previsione a breve termine.

6 LISTA PUBBLICAZIONI

6.1 Lista pubblicazioni tratte dalla tesi di Dottorato

- (1) **G. De Luca**, M. Gallo (2020) *The use of Artificial Neural Networks for extending road traffic monitoring data spatially: an application to the neighbourhoods of Benevento*. Transportation Research Procedia, 45, pp. 635-642, ISSN: 2352-1465, doi: 10.1016/j.trpro.2020.03.047.
- (2) M. Gallo, **G. De Luca**, L. D’Acerno, M. Botte (2019) *Artificial Neural Networks for Forecasting Passenger Flows on Metro Lines*. Sensors, 19, article id. 3424, 14 pages, ISSN: 1424-8220, doi: 10.3390/s19153424.
- (3) M. Gallo, **G. De Luca** (2018) *Spatial extension of road traffic sensor data with Artificial Neural Networks*. Sensors, 18, article id. 2640, 14 pages, ISSN: 1424-8220, doi: 10.3390/s18082640.
- (4) **G. De Luca**, M. Gallo (2017) *Artificial Neural Networks for forecasting user flows in transportation networks: literature review, limits, potentialities and open challenges*. Proceedings of 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, Naples, Italy, 26-28 June 2017, pp. 919-923, ISBN: 978-1-5090-6484-7, doi: 10.1109/MTITS.2017.8005644.

6.2 Altre pubblicazioni

- (5) R. Iannella, M. Gallo, **G. De Luca**, F. Simonelli (2017) *Regression Methods for Spatially Extending Traffic Data*. International Journal of Engineering Technology, Management and Applied Sciences, 5, pp. 5-14, ISSN: 2349-4476.
- (6) M. Gallo, F. Simonelli, **G. De Luca** (2017) *The potential of energy-efficient driving profiles on railway consumption: a parametric approach*. In “Transport Infrastructure and Systems” (Editors: G. Dell’Acqua, F. Wegman), Taylor & Francis Group, pp. 819-824, ISBN: 978-1-138-09009-1, doi: 10.1201/9781315281896-105.
- (7) M. Gallo, **G. De Luca** (2016) *A methodology for evaluating urban traffic plan scenarios from the point of view of traffic noise*. WSEAS Transactions on Circuits and Systems, 15, pp. 187-197, ISSN: 1109-2734, E-ISSN: 2224-266X.
- (8) M. Gallo, **G. De Luca** (2016) *Evaluation of urban traffic plan scenarios in terms of traffic noise abatement*. International Journal of Transportation Systems, 1, pp. 98-103, ISSN: 2367-9131.
- (9) M. Gallo, F. Simonelli, **G. De Luca**, V. De Martinis (2015) *Estimating the effects of energy-efficient driving profiles on railway consumption*. Proceedings of IEEE EEEIC 2015 – 15th International Conference on Environment and Electrical Engineering, Rome, Italy, June 2015, pp. 813-818, ISBN: 978-1-4799-7992-9, doi: 10.1109/EEEIC.2015.7165269.
- (10) M. Gallo, F. Simonelli, **G. De Luca**, V. De Martinis (2015) *Il risparmio energetico per il trasporto ferroviario: le potenzialità dell’eco-drive*. SmartCity & MobilityLab, n. 7, pp. 14-19, ISSN 2281-6534.

- (11) G. Corapi, V. De Martinis, A. Placido, **G. De Luca** (2014) *Impacts of energy saving strategies (ESSs) on rail services and related effects on travel demand*. In “Computers in Railways XIV” (Editors: C.A. Brebbia, N. Tomii, P. Tzieropolu, J.M. Mera), WIT Transactions on the Built Environment, vol. 135, WIT Press, Southampton, United Kingdom, pp. 709-720, ISSN: 1743-3541, ISBN: 978-1-78466-023-9, doi: 10.2495/SC140491.
- (12) M. Gallo, **G. De Luca**, V. De Martinis (2014) *The effects of urban traffic plans on noise abatement: a case study*. In “The Sustainable City IX – Urban Regeneration and Sustainability” (Editors: N. Marchettini, C.A. Brebbia, R. Pulselli, S. Bastianoni), WIT Transactions on Ecology and Environment Series, vol. 191, WIT Press, Southampton, United Kingdom, pp. 583-594, ISSN: 1743-3509, doi: 10.2495/CR140591.
- (13) M. Gallo, **G. De Luca** (2014) *Piani Urbani del Traffico. La valutazione degli scenari ai fini della riduzione dell'inquinamento acustico*. SmartCity & MobilityLab, n. 4, pp. 4-10, ISSN 2281-6534.
- (14) M. Gallo, **G. De Luca**, L. D’Acierno (2014) *The design of signalised intersections at area level. Models and Methods*. TEMA, Special Issue June 2014, pp. 453-464, ISSN: 1970-9870.
- (15) L. D’Acierno, **G. De Luca**, M. Gallo (2013) *Minimisation of total delay in two-way coordinated arterials*. In “Urban Transport XIX – Urban Transport and the Environment in the 21st Century” (Editor: C.A. Brebbia), WIT Transactions on the Built Environment Series, vol. 130, WIT Press, Southampton, United Kingdom, pp. 41-51, ISSN: 1746-4498, ISBN: 978-1-84564-716-2, doi: 10.2495/UT130041.
- (16) M. Gallo, F. Simonelli, **G. De Luca** (2013) *Progetto della regolazione di intersezioni semaforizzate a livello di area: metodologie*. In “Linee guida per l’analisi e il progetto di reti urbane di trasporto. Metodologie: stato dell’arte e modelli” (a cura di A. Vitetta), Maggioli Editore, pp. 163-181, ISBN: 978-88-387-8220-6.
- (17) M. Gallo, **G. De Luca** (2013) *Descrizione del caso di studio: il sistema stradale urbano del Comune di Benevento*. In “Linee guida per l’analisi e il progetto di reti urbane di trasporto. Applicazioni” (a cura di A. Vitetta), Maggioli Editore, pp. 13-25, ISBN: 978-88-387-8241-1.
- (18) M. Gallo, **G. De Luca**, L. D’Acierno (2013) *Progetto della regolazione di intersezioni semaforizzate a livello di area: applicazione*. In “Linee guida per l’analisi e il progetto di reti urbane di trasporto. Applicazioni” (a cura di A. Vitetta), Maggioli Editore, pp. 93-110, ISBN: 978-88-387-8241-1.