

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA · SEDE DI CESENA

CORSO DI LAUREA IN
INGEGNERIA E SCIENZE INFORMATICHE

RELAZIONE ATTIVITÀ DI TIROCINIO

SVOLTO PRESSO:

ALEXIDE S.R.L.

CESENA(FC)

Via Ilaria Alpi 65

47522

SPERIMENTAZIONE DI TECNICHE DI
MACHINE LEARNING DI SUPPORTO A
SISTEMI MRP (MANUFACTURING
RESOURCE PLANNING)

TUTOR AZIENDALE:
ALESSANDRO FRESCHI

TUTOR ACCADEMICO:
MATTEO FERRARA

PRESENTATA DA:
MICHELE PIO PRENCIPE

ANNO ACCADEMICO: 2020/2021

PERIODO DI SVOLGIMENTO: 13/04/2021 - 17/05/2021

Indice

1	Introduzione	2
2	Tecnologie Utilizzate	3
3	Attività	4
3.1	Pulp, NumPy e Pandas	4
3.2	scikit-learn	5
3.3	TensorFlow	5
3.4	Preparazione Dati	6
3.5	Rete Neurale Finale	9
4	Conclusioni	12

1. Introduzione

Ho svolto l'attività di tirocinio presso l'azienda Alexide S.r.l., Cesena(FC), 47522, in modalità "smart-working", da casa, a causa emergenza sanitaria Covid-19, nel periodo compreso tra il 13/04/2021 e il 17/05/2021 per una durata complessiva di 150 ore, corrispondenti a 6 CFU.

Alexide srl è una società nata nel 2005 che sviluppa software in ambito documentale, configurazione di prodotto, CAD, PDM, PLM, Produzione, IoT, AI, Big Data. In particolare, lavorano al progetto chiamato SolidRules, una proposta software divenuta riferimento per il settore manifatturiero. Lavora in team utilizzando i paradigmi dell'Agile Programming mentre i principali linguaggi utilizzati sono C# e Python. Sta lavorando per portare le tecniche di AI e Machine Learning nella quotidianità.

In questo ambito aziendale sono stato introdotto nella gestione di produzione MRP(Manufacturing Resource Planning). MRP è una tecnica di pianificazione delle risorse aziendali e si occupa di fare un'analisi dei dati degli ordini e dei cicli di produzione. Nello specifico, sulla base di dati storici di produzione prelevati da un campione di aziende si andranno ad analizzare i tempi di produzione e di approvvigionamento di determinati articoli con lo scopo di prevedere tempi di approvvigionamento e di produzione su articoli simili. Il campione di dati per ogni articolo già prodotto fornisce un elenco di dati anagrafici che lo caratterizzano e i tempi storici di approvvigionamento dei materiali e delle singole fasi di lavoro.

L'obiettivo del tirocinio è, quindi, quello di dedurre i tempi ciclo e i tempi di approvvigionamento di un articolo mai prodotto sulla base dei dati storici di articoli simili, tramite l'addestramento di una rete neurale costruita a partire dalla libreria TensorFlow di Python. Lo sviluppo del modello di rete, tuttavia è stato preceduto da una fase di studio, esercitazioni e progettazione anche attraverso l'uso di altre librerie come Pulp, Pandas, NumPy, scikit-learn e la stessa TensorFlow, propedeutiche allo sviluppo del modello e utili per poter preparare i dati da dare in input alla rete neurale nella maniera ottimale.

2. Tecnologie Utilizzate

Durante lo svolgimento mi sono imbattuto in varie tecnologie per gestire anche vari programmi in modo organizzato e veloce. In particolare ho impiegato le seguenti tecnologie:

- Linguaggi di programmazione:
 - Python, linguaggio orientato agli oggetti, adatto, tra gli altri usi, alla computazione numerica e utile per via della grande varietà di librerie da poter consultare.
 - Utilizzo di Excel per la manipolazione di dati
- Librerie:
 - Pulp, per la modellazione di problemi di Programmazione Lineare
 - NumPy, per la manipolazioni di matrici di dati
 - Pandas, per la manipolazioni di dati attraverso Series e DataFrame
 - scikit-learn, per il Machine Learning
 - TensorFlow, per la costruzione di un modello di rete neurale
- Piattaforme:
 - Anaconda come piattaforma, particolarmente adatta alla Data Science
 - Spyder come IDE di programmazione
- Sistemi Operativi:
 - Windows10

3. Attività

Una parte fondamentale che mi ha permesso di svolgere i compiti assegnati dall'azienda è stata quella di studiare attentamente tutta una serie di tecniche e librerie propedeutiche alla realizzazione del modello di rete finale.

3.1 Pulp, NumPy e Pandas

Lo studio della libreria Pulp mi ha permesso di entrare nell'ottica del MRP attraverso il livello più basso, ovvero la Programmazione Lineare. La Programmazione Lineare è quella branca della ricerca operativa che si occupa di studiare algoritmi di risoluzione per problemi di ottimizzazione a livello lineare. In questo frangente mi sono limitato a fare qualche esercizio di PL e iniziare a prendere confidenza con i software suggeriti, come Anaconda e Spyder IDE. In un secondo momento ho studiato come funziona la libreria NumPy, utile per la manipolazione dei dati attraverso le matrici, ma la libreria senz'altro più utile per tale compito è stata Pandas, con la quale è possibile trasformare tabelle vere e proprie con grandi quantità di dati in un oggetto chiamato DataFrame. Anche in questa fase mi sono limitato allo studio approfondito delle librerie e ho cominciato a fare qualche esercizio con piccole quantità di dati, sia casuali che presi DataSet noti.

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Age
0	0.435	0.335	0.110	0.334	0.1355	0.0775	0.0965	7
1	0.585	0.450	0.125	0.874	0.3545	0.2075	0.2250	6
2	0.655	0.510	0.160	1.092	0.3960	0.2825	0.3700	14
3	0.545	0.425	0.125	0.768	0.2940	0.1495	0.2600	16
4	0.545	0.420	0.130	0.879	0.3740	0.1695	0.2300	13

Figura 3.1: Esempio DataFrame

3.2 scikit-learn

Per quanto riguarda il Machine Learning ho impiegato la libreria scikit-learn, utile tra le altre cose per la classificazione e la regressione. In questa fase per misurare la correlazione di dati ho utilizzato anche la libreria matplotlib.

Con la semplice regressione lineare si è poi costruito un modello di Machine Learning che impara e si addestra sui dati con una precisione di circa il 50%.

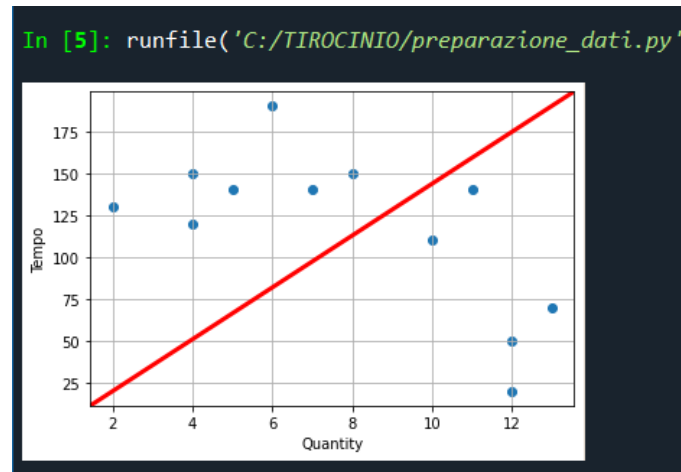


Figura 3.2: plot della correlazione tra quantità dei figli e tempo con linea di approssimazione dei dati

3.3 TensorFlow

Con lo studio della libreria TensorFlow ho imparato il funzionamento di una rete neurale e del Machine Learning in generale. Grazie a `keras.Sequential` viene addestrato un modello sequenziale in cui ogni nodo elabora dati in input tramite la cosiddetta funzione "relu" e produce un output da dare in pasto ai nodi successivi. Il modello itera in varie epoche passate dal programmatore fino ad arrivare a predizioni sempre più vicine a quelle reali.

```

Epoch 2/100
1049/1049 [=====] - 1s 514us/step - loss: 65955.2233
Epoch 3/100
1049/1049 [=====] - 1s 559us/step - loss: 1408.7874
Epoch 4/100
1049/1049 [=====] - 1s 553us/step - loss: 20834.5202
Epoch 5/100
1049/1049 [=====] - 1s 553us/step - loss: 4402.4451
Epoch 6/100
1049/1049 [=====] - 1s 544us/step - loss: 1460.0521
Epoch 7/100
1049/1049 [=====] - 1s 544us/step - loss: 184.1220
Epoch 8/100
1049/1049 [=====] - 1s 538us/step - loss: 2982.0353
Epoch 9/100
1049/1049 [=====] - 1s 574us/step - loss: 23600.3168
Epoch 10/100
1049/1049 [=====] - 1s 559us/step - loss: 1894.6387
Epoch 11/100
1049/1049 [=====] - 1s 529us/step - loss: 2630.3255
Epoch 12/100
1049/1049 [=====] - 1s 559us/step - loss: 15492.5101

```

Figura 3.3: Addestramento del modello sequenziale di rete

3.4 Preparazione Dati

Per la preparazione dei dati ho impiegato la libreria Pandas in modo da manipolare le varie tabelle da fornire in input alla rete neurale. Le tabelle di dati iniziali sono le classiche usate nel MRP:

- Anagrafiche
- Distinte
- Oggetti Padre(ad esempio Cassettiera)
- Oggetti Figli(as esempio Maniglia)
- Tempi di Produzione

Da queste tabelle occorre individuare un DataFrame unico per scoprire se c'è correlazione, ad esempio tra quantità dei figli e tempi di produzione oppure tra la descrizione degli oggetti e i tempi di lavorazione. Si noti che è stato volontariamente evitato di applicare un text processing o NLP(Natural Language Processing) in quanto potrebbero essere oggetto di studi o di miglioramenti futuri.

	A	B	C
1	ANAGRAFICHE		
2	Codice	Tipologia	Modello
3	FIANCO01	FIANCO	LUX
4	FIANCO02	FIANCO	STANDARD
5	MANIGLIA004	MANIGLIA	STANDARD
6	MANIGLIA005	MANIGLIA	STANDARD
7	TOP01	TOP	LUX
8	TOP01	TOP	STANDARD
9	CASSETTIERA0001	CASSETTIERA	ESA
10	CASSETTIERA0002	CASSETTIERA	POA
11	MANIGLIA006	MANIGLIA	LUX
12	FIANCO03	FIANCO	STANDARD
13	TOP02	TOP	STANDARD
14	MANIGLIA007	MANIGLIA	STANDARD
15	FIANCO04	FIANCO	LUX
16	CASSETTIERA0003	CASSETTIERA	ESA
17	CASSETTIERA0004	CASSETTIERA	POA
18	CASSETTIERA0005	CASSETTIERA	ESA
19	CASSETTIERA0006	CASSETTIERA	POA

Figura 3.4: Esempio Anagrafiche

```

Console 1/A
Out[7]:
  Codice Padre  Tipologia_Padre  ... Qty_TOP_STANDARD  Tempo
0  CASSETTIERA0001  CASSETTIERA  ...              1.0   140.0
1  CASSETTIERA0002  CASSETTIERA  ...              1.0   150.0
2  CASSETTIERA0003  CASSETTIERA  ...              0.0   120.0
3  CASSETTIERA0004  CASSETTIERA  ...             14.0   140.0
4  CASSETTIERA0005  CASSETTIERA  ...              0.0    70.0
5  CASSETTIERA0006  CASSETTIERA  ...              0.0    20.0
6  CASSETTIERA0007  CASSETTIERA  ...              7.0   190.0
7  CASSETTIERA0008  CASSETTIERA  ...              0.0   130.0
8  CASSETTIERA0009  CASSETTIERA  ...              0.0   140.0
9  CASSETTIERA0010  CASSETTIERA  ...              0.0   150.0
10 CASSETTIERA0011  CASSETTIERA  ...              0.0    50.0
11 CASSETTIERA0012  CASSETTIERA  ...              0.0   110.0

[12 rows x 10 columns]

In [8]: df.columns
Out[8]:
Index(['Codice Padre', 'Tipologia_Padre', 'Modello_Padre', 'Qty_FIANCO_LUX',
      'Qty_FIANCO_STANDARD', 'Qty_MANIGLIA_LUX', 'Qty_MANIGLIA_STANDARD',
      'Qty_TOP_LUX', 'Qty_TOP_STANDARD', 'Tempo'],
      dtype='object', name='columns')

```

Figura 3.5: Dataframe rapporto quantità figli e padri

La descrizione dell'oggetto figlio viene, quindi, suddivisa in lettere. Ogni lettera viene poi moltiplicata per la quantità dei figli.

Ne consegue un dataframe in cui in ogni riga ci sono le quantità di lettere totali di ogni padre e il suo tempo effettivo di lavorazione. Il dataframe per l'addestramento presenta, quindi le seguenti statistiche:

```
In [4]: df
Out[4]:
```

	A	B	C	D	E	F	...	V	W	X	Y	Z	t_eff
0	25.0	5.0	11.0	1.0	8.0	7.0	...	4.0	0.0	2.0	0.0	0.0	0.000000
1	10.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	0.000000
2	12.0	0.0	1.0	0.0	0.0	0.0	...	0.0	4.0	0.0	0.0	0.0	2.896667
3	13.0	1.0	8.0	2.0	7.0	5.0	...	2.0	0.0	4.0	0.0	2.0	0.000000
4	71.0	14.0	22.0	15.0	20.0	26.0	...	3.0	7.0	2.0	0.0	6.0	0.000000
...
1222	29.0	0.0	1.0	7.0	34.0	15.0	...	7.0	2.0	15.0	0.0	0.0	4.210000
1223	22.0	0.0	1.0	4.0	21.0	12.0	...	4.0	0.0	12.0	0.0	0.0	4.210000
1224	33.0	14.0	0.0	51.0	10.0	0.0	...	10.0	14.0	0.0	0.0	0.0	3.570000
1225	27.0	5.0	0.0	6.0	1.0	0.0	...	1.0	3.0	0.0	0.0	8.0	3.570000
1226	14.0	3.0	0.0	4.0	1.0	0.0	...	1.0	2.0	0.0	0.0	4.0	3.570000

[1227 rows x 26 columns]

```
In [5]: df.describe()
Out[5]:
```

	A	B	...	Z	t_eff
count	1227.000000	1227.000000	...	1227.000000	1227.000000
mean	152.938060	25.022005	...	5.821516	5.470481
std	501.884174	110.554728	...	16.883439	9.163686
min	0.000000	0.000000	...	0.000000	0.000000
25%	7.000000	0.000000	...	0.000000	0.420000
50%	35.000000	3.000000	...	0.000000	1.736667
75%	126.000000	16.000000	...	4.000000	6.123571
max	13444.000000	2128.000000	...	248.000000	76.720000

[8 rows x 26 columns]

Figura 3.6: Dataset finale e statistiche

Tuttavia, non tutto il dataset è stato utilizzato per l'addestramento del modello. In particolare, i 2/3 dei dati sono stati utilizzati per il training del modello, mentre il restante 1/3 per la validazione. Di seguito una tabella con le cardinalità dei vari set di dati:

	Righe	Colonne
Dataset	1227	26
Training Set	818	26
Validation Set	409	26

3.5 Rete Neurale Finale

Infine, la rete finale presenta una struttura su 3 livelli: uno di input composto dalle 25 colonne corrispondenti alle features del set di dati, 1 livello che riceve in ingresso i dati di input ed infine un livello di output composto da un solo nodo corrispondente all'unica colonna di label. La funzione di attivazione di ogni nodo è la cosiddetta "ReLU" usata di default in TensorFlow. Per la loss function è stato utilizzato il classico MSE (Mean Squared Error) e come ottimizzatore "adam", la batch_size è di 1 campione alla volta. Il modello è stato, poi, addestrato su un set di epoche: 1, 10, 100, 1000 scoprendo che 100 fosse il numero di epoche più adatto. Infatti con un 1 o 10 il modello non veniva addestrato correttamente. La differenza in termini di MSE tra 100 e 1000 era minima, tuttavia con 1000 epoche il tempo di calcolo era estremamente superiore.

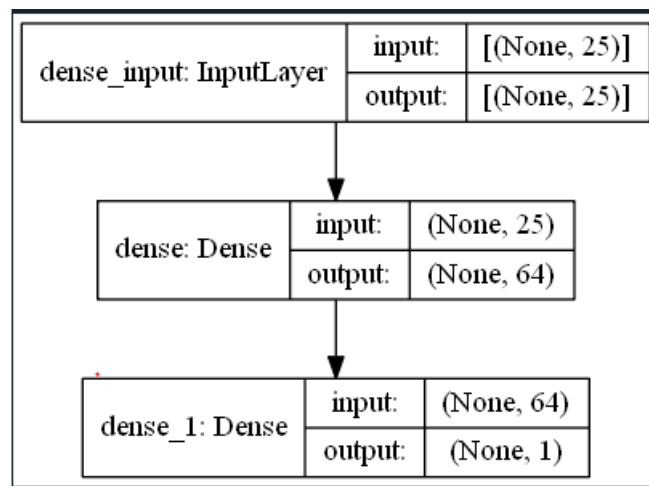


Figura 3.7: Struttura della rete finale

Viene, quindi, addestrato un modello sui dati in cui si cerca di scoprire se vi è correlazione tra le lettere presenti nella descrizione dei figli e i tempi di produzione. Si scopre che il modello, inizialmente predice in maniera errata per poi avvicinarsi sempre di più al tempo effettivo di produzione.

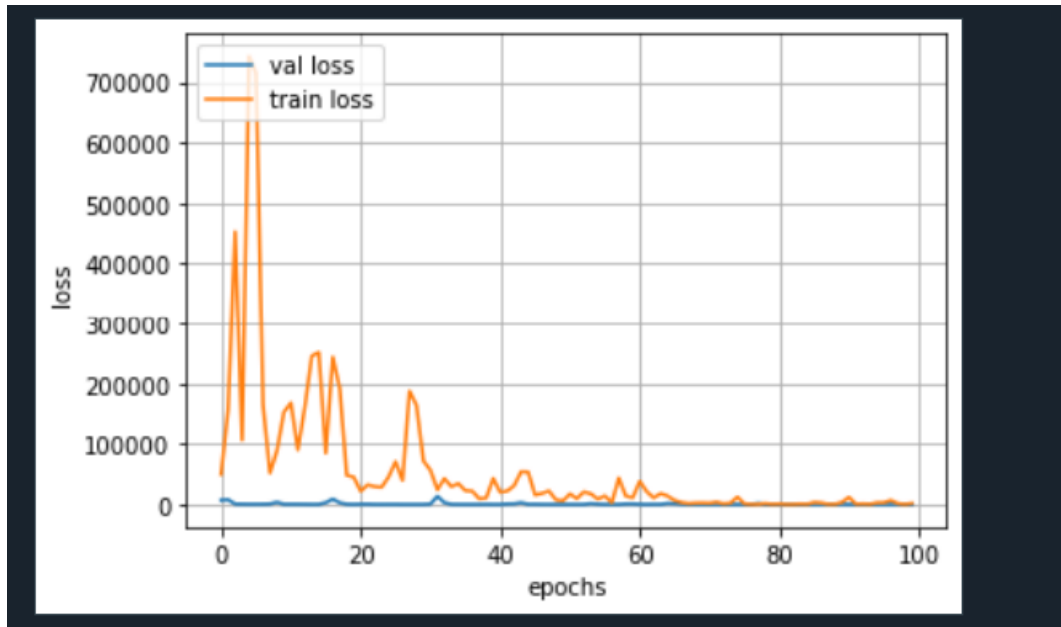


Figura 3.8: Funzione di loss sul training set e validation set

A	0.898679
B	3.110739
C	5.149330
D	0.460643
E	-11.454941
F	11.609435
G	-5.353258
H	-15.477146
I	2.617255

Figura 3.9: Coefficiente che indica quanto ogni lettera influisce sul tempo finale di produzione

Essendo un problema di regressione, l'accuratezza del modello si può misurare sui dati di training in relazione ai dati reali attraverso MSE (Mean Squared Error) o attraverso il coefficiente R^2 , un numero compreso tra 0 e 1 che misura una proporzione tra la variabilità dei dati e la correttezza del modello, più si avvicina a 1 più è corretto il modello.

```
In [45]: runfile('C:/TIROCINIO/dati/reteNeurale.py', wdir='C:/TIROCINIO/dati')
Mean squared error: 20.891
R-squared coefficient: 0.7238
```

Figura 3.10: MSE e R^2 del training del modello rispetto ai dati reali

4. Conclusioni

Mi ritengo pienamente soddisfatto del percorso e dell'esperienza lavorativa all'interno dell'azienda Alexide S.r.l., in cui ho imparato ad utilizzare nuove tecnologie che fino a prima del tirocinio mi erano sconosciute. Quello del MRP e del Machine Learning, in generale è un ambito poco conosciuto, ma di cui mi sono appassionato fin da subito.

Durante tutto il mio percorso all'interno dell'azienda mi sono sempre relazionato con il tutor aziendale Alessandro Freschi, il quale mi ha seguito durante tutte le fasi di studio, analisi, esercitazione, progettazione e sviluppo.

Grazie, quindi, al sostegno di Alessandro Freschi e alla passione per la materia, le difficoltà incontrate durante il percorso si sono rivelate minime.

Nel corso di queste esperienze sono riuscito ad imparare nozioni importanti nell'ambito del MRP, del Machine Learning, dell'Analisi di Dati e su come questi si applicano e quali tecnologie utilizzano.

Un difetto che posso trovare a questa esperienza è che, lavorando da casa non ho avuto la possibilità di vivere a pieno il modo di lavorare dell'azienda.