



Autoencoder e Predittore di Stati

Presentazione del progetto per il corso di Deep Learning, A.A. 2021/22

Luca Lavazza, Francesco Rossi

Introduzione



- Autoencoder
- Predittore di stati
 - 4 a 1
 - 6 a 1

Indicazioni Generali - Suddivisione dei Dataset

- **Task 1:**

- *Training set:* 9.2×10^5 vettori
- *Validation set:* 2.31×10^5 vettori
- *Test set:* 2.31×10^5 vettori

- **Task 2:**

- *Training set:*
 - *train_x:* $(1.54 \times 10^5, 4, 340)$
 - *train_y:* $(1.54 \times 10^5, 1, 340)$
- *Validation set:*
 - *val_x:* $(5.1 \times 10^4, 4, 340)$
 - *val_y:* $(5.1 \times 10^4, 1, 340)$
- *Test set:*
 - *test_x:* $(5.1 \times 10^4, 4, 340)$
 - *test_y:* $(5.1 \times 10^4, 1, 340)$
- Sono stati scartati circa 10^5 vettori per far tornare i conti della suddivisione

- **Task 3:**

- *Training set:*
 - *train_x:* $(10^5, 6, 340)$
 - *train_y:* $(10^5, 1, 340)$
- *Validation set:*
 - *val_x:* $(3.35 \times 10^4, 4, 340)$
 - *val_y:* $(3.35 \times 10^4, 1, 340)$
- *Test set:*
 - *test_x:* $(3.35 \times 10^4, 4, 340)$
 - *test_y:* $(3.35 \times 10^4, 1, 340)$
- Sono stati scartati circa 2×10^5 vettori per far tornare i conti della suddivisione

Il dataset di partenza era composto da 1.382×10^6 vettori di 340 valori binari ciascuno, estremamente sparsi (pochissimi 1 in confronto agli 0).

Indicazioni Generali - Metriche



- Precision



Circa 1

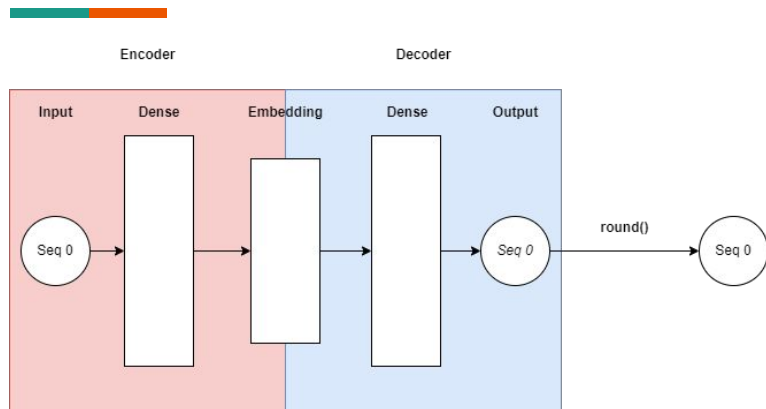
- Recall

- Loss



Sotto 0.01

Task 1: Autoencoder classico - Struttura



- 3 livelli interni di attivazione (embedding compresso)
- Attivazione: ReLU
- Ottimizzazione: Adam e Nadam
- Regularizzazione: Early Stopping e Batch Normalization sempre utilizzate.

Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Accuratezza
340	170	48	170	340	69.807%
340	170	65	170	340	92.319%
340	170	80	170	340	99.755%

Task 1: Autoencoder classico - Modello 2

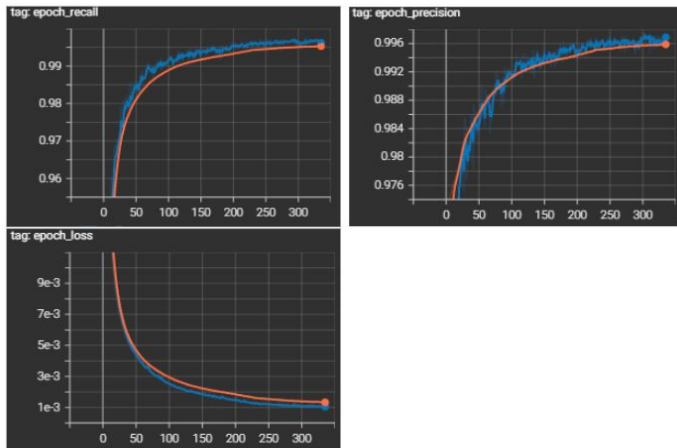
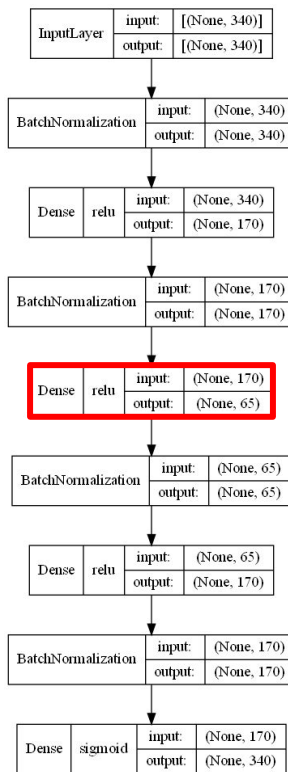


Figure 4: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



Parametri	Tipo	Valori
Kernel initializer	heunifom	-
Ottimizzatore	Nadam	$lr=0.0007$, $\beta_{0.1}=0.95$, $\beta_{0.2}=0.999$
Kernel regularizer	-	-
Batch size	-	5000

*% di accuratezza di ricostruzione
sul testset: 92.3%*

Task 1: Autoencoder classico - Modello 3

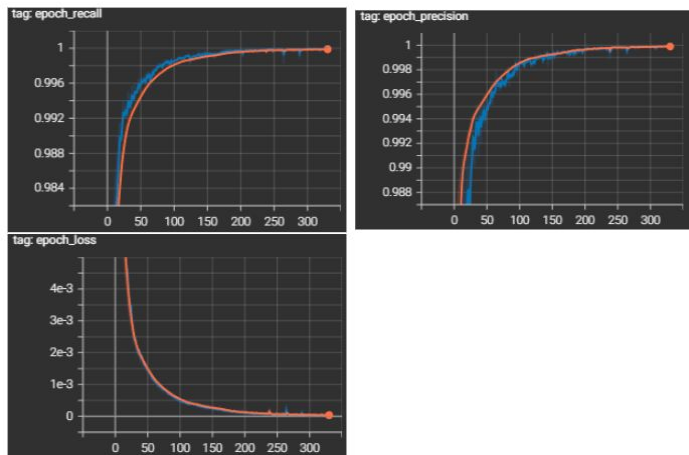
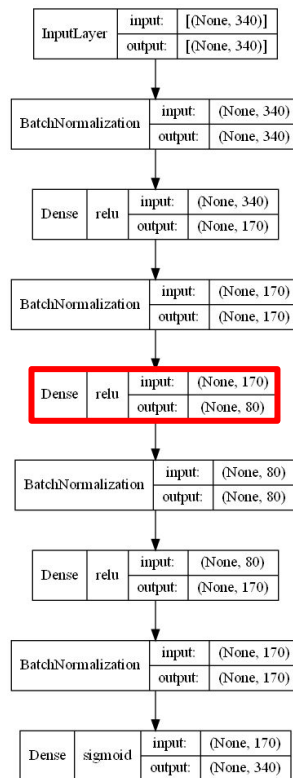


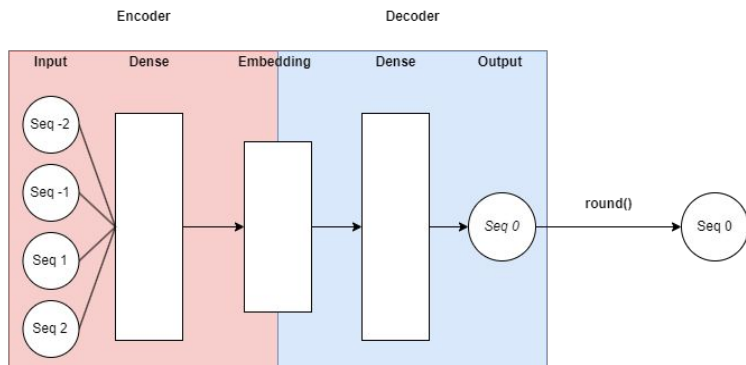
Figure 6: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



Parametri	Tipo	Valori
Kernel initializer	he_uniform	-
Ottimizzatore	Nadam	$lr=0.001, beta_1=0.9, beta_2=0.999$
Kernel regularizer	-	-
Batch size	-	5000

% di accuratezza di ricostruzione
sul testset: 99.75%

Task 2 - Predittore 4 a 1 - Struttura



- 5 livelli interni di attivazione (embedding compreso)
- Attivazione: ReLU
- Ottimizzazione: Adam e Nadam
- Regularizzazione: Early Stopping e Batch Normalization sempre utilizzate.

Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Accuratezza
(340*4)1360	→ 1340	→ 268	→ 134	→ 268	→ 1340	→ 340	-----▷	92.512%
(340*4)1360	→ 670	→ 134	→ 67	→ 134	→ 670	→ 340	-----▷	90.681%

Task 2 - Predittore 4 a 1 - Modello 1

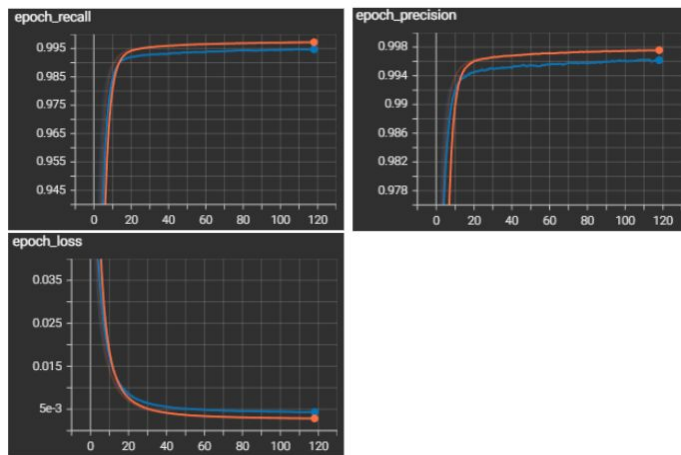
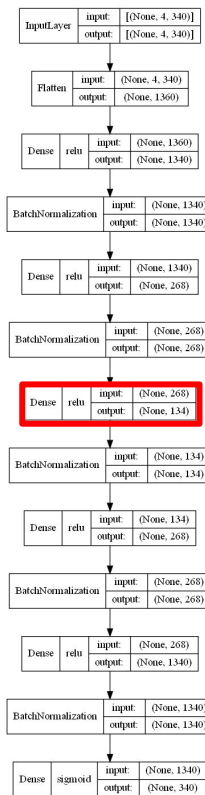


Figure 10: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



Parametri	Tipo	Valori
Kernel initializer	he_uniform	-
Ottimizzatore	Adam	$lr=7 \cdot 10^{-5}$
Kernel regularizer	L2	$5 \cdot 10^{-6}$
Batch size	-	50

*% di accuratezza di ricostruzione
sul testset: 92.5%*

Task 2 - Predittore 4 a 1 - Modello 2

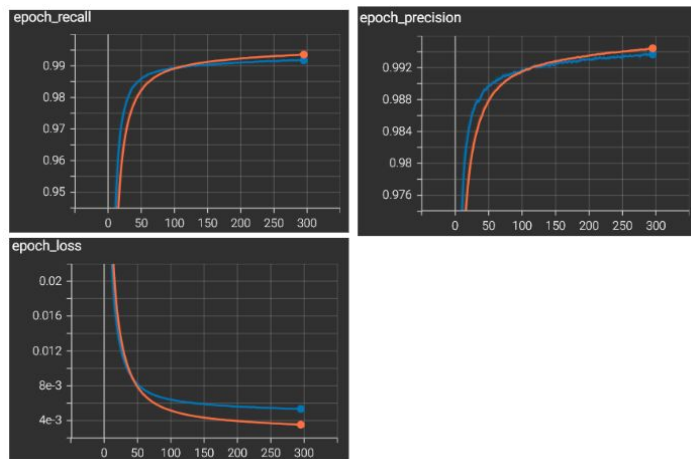
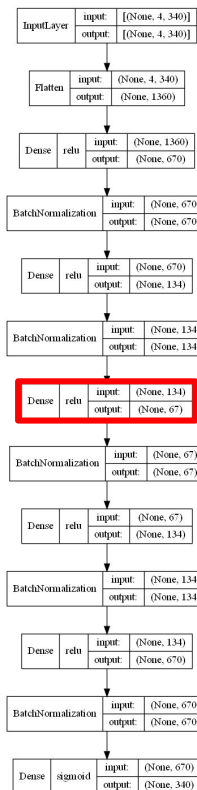


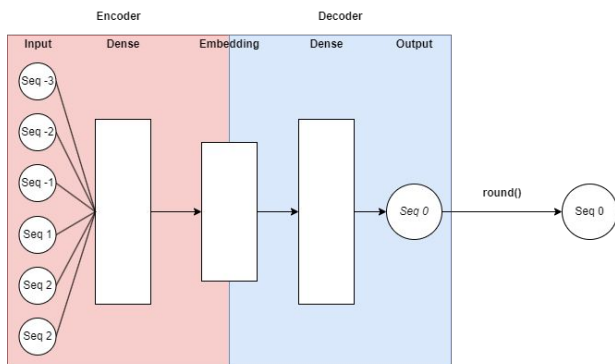
Figure 12: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



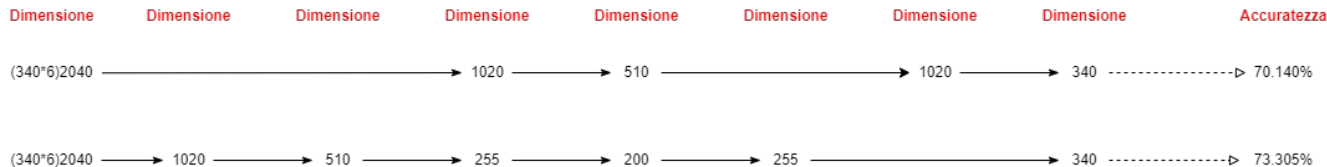
Parametri	Tipo	Valori
Kernel initializer	he_uniform	-
Ottimizzatore	Adam	$lr=7 \cdot 10^{-5}$
Kernel regularizer	L2	$5 \cdot 10^{-6}$
Batch size	-	50

% di accuratezza di ricostruzione
sul testset: 90.6%

Task 3 - Predittore 6 a 1 - Struttura



- 3 livelli interni di attivazione (embedding compreso) nel primo caso, 5 nel secondo
- Attivazione: ReLU
- Ottimizzazione: Adam e Nadam
- Regularizzazione: Early Stopping e Batch Normalization sempre utilizzate.



Task 3 - Predittore 6 a 1 - Modello 1

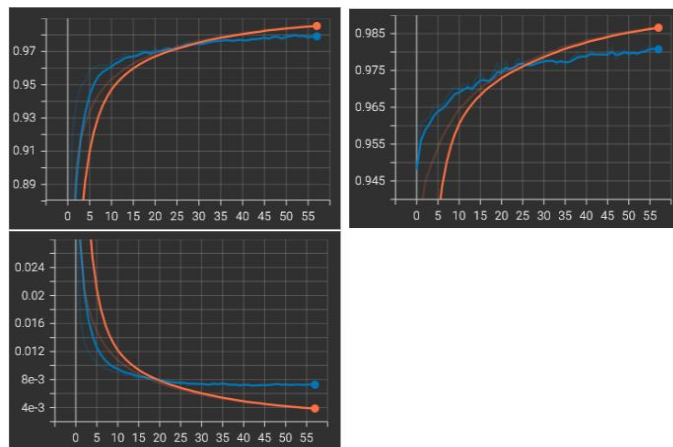
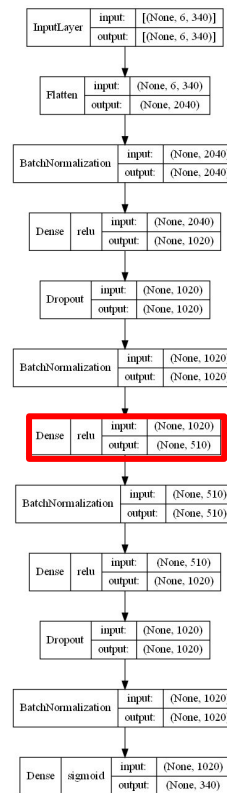


Figure 16: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



Parametri	Tipo	Valori
Kernel initializer	he_normal	-
Optimizers	Nadam	$lr = 0.009, beta_1 = 0.95$
Kernel regularizer	L2	$1e-10$
Batch size	-	400

% di accuratezza di ricostruzione
sul testset: 70.1%

Task 3 - Predittore 6 a 1 - Modello 2

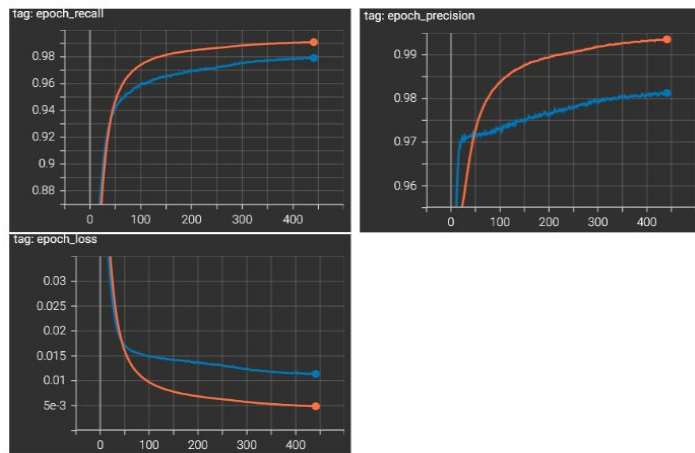
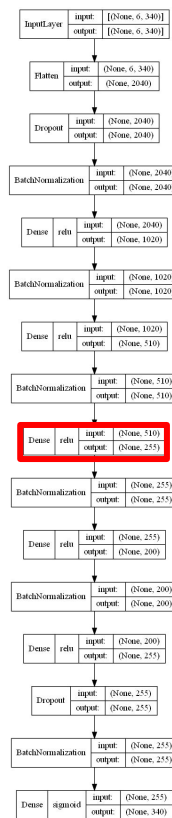


Figure 18: Andamento di recall, precision e loss su **train** e **validation** sulla base delle epoche.



Parametri	Tipo	Valori
Kernel initializer	he_normal	-
Optimizer	adam	$lr = 0.0003700088651332496$
Kernel regularizer	L2	$1.967657265571901e - 06$
Dropout	-	0.05 entrambi
Batch size	-	1000

*% di accuratezza di ricostruzione
sul testset: 73.3%*

Recap dei Task e Relativi Risultati



- **Task 1: autoencoder tradizionale**

In generale ottimi risultati, fino ad una riduzione dell'input di 5 volte (% di vettori ricostruiti esatti >90% sul testset). Oltre, l'accuratezza nella ricostruzione diminuiva drasticamente al diminuire della dimensione dell'embedding.

- **Task 2: predittore con struttura 4 in-1 out**

Le reti illustrate hanno ottenuto risultati soddisfacenti, con un valore di accuratezza di predizione sul test set >90%.

- **Task 3: predittore con struttura 6 in-1 out**

Reti più complesse, task più complesso che ha prodotto risultati inferiori al caso precedente (circa 70% sul testset). Risultati influenzati dall'alto overfitting e dall'elevata sparsità dei dati.