Autoencoder e Predittore di Stati

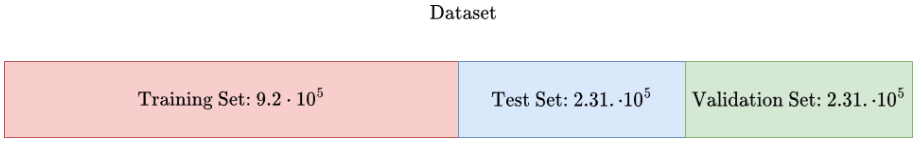
Presentazione del progetto per il corso di Deep Learning, A.A. 2021/22

Luca Lavazza, Francesco Rossi

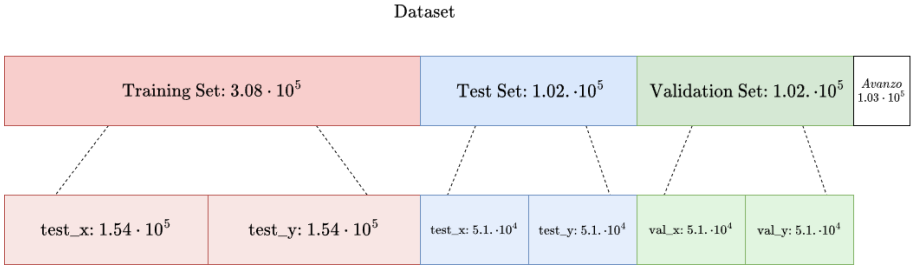
Introduzione

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

Indicazioni Generali - Suddivisione dei Dataset

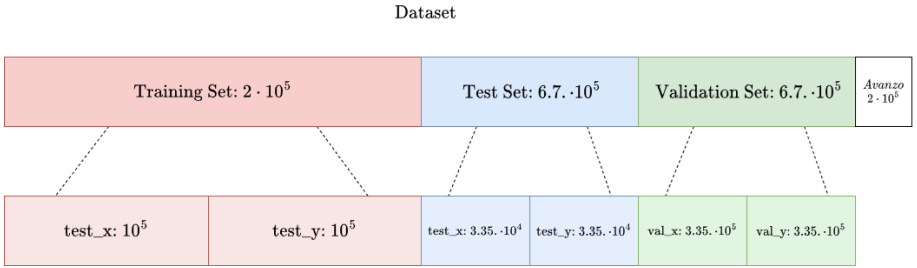


Task 1



Task 2

Un vettore base è costituito da 340 valori binari, e risulta essere sempre molto sparso (pochissimi 1 in confronto agli 0).



Task 3

Indicazioni Generali - Metriche

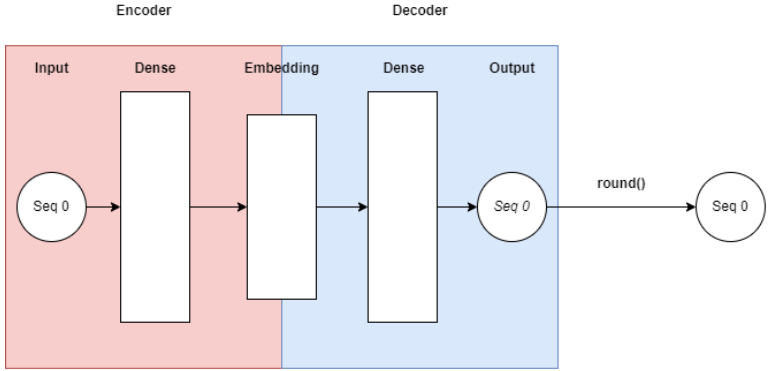
- Precision
- Recall

Circa 1

- Loss

Sotto 0.01

Task 1: Autoencoder classico - Struttura



- 3 livelli interni totali (embedding compreso)
- Attivazione: ReLU
- Ottimizzazione: Adam e Nadam
- Regolarizzazione: 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 1

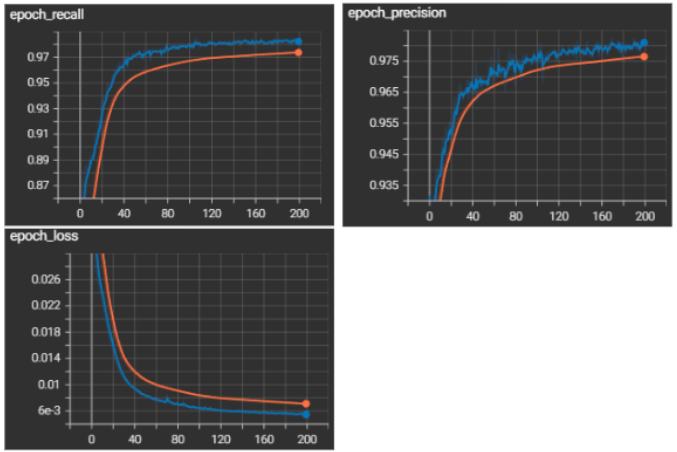
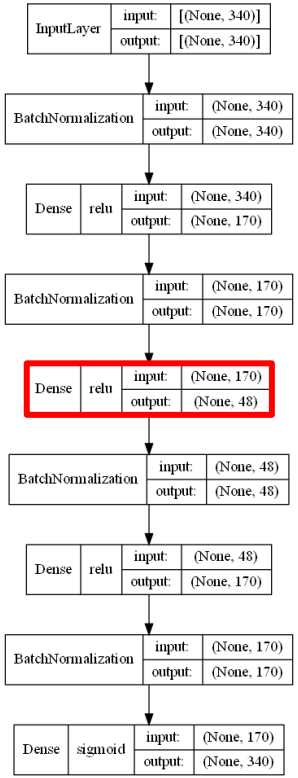


Figure 2: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	he_uniform	-
Ottimizatore	Adam	lr=0.0001
Kernel regulizer	L2	10^{-5}
Batch size	-	150

Task 1: Autoencoder classico - Modello 2

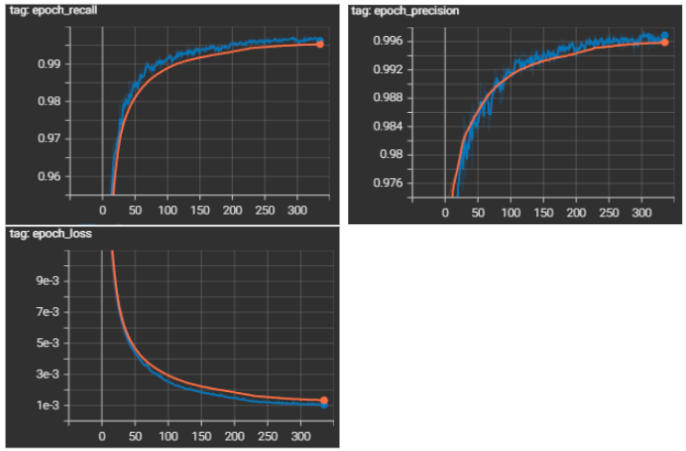
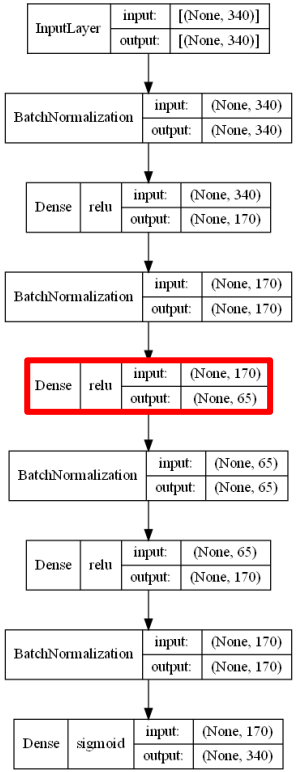


Figure 4: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	heuniform	-
Ottimizzatore	Nadam	lr=0.0007, $beta\_1$=0.95, $beta\_2$=0.999
Kernel regulizer	-	-
Batch size	-	5000

Task 1: Autoencoder classico - Modello 3

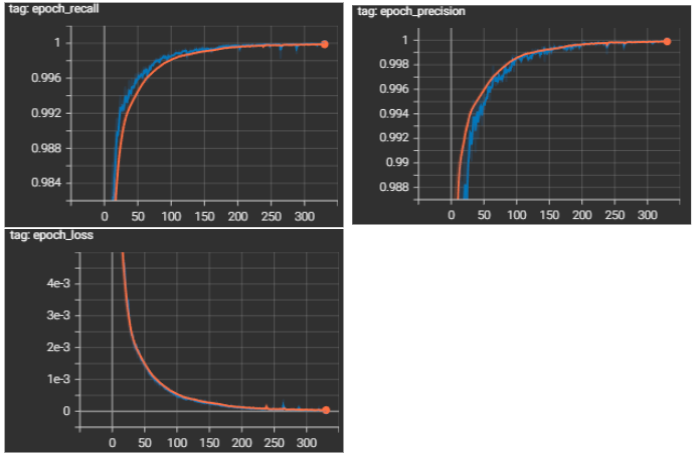
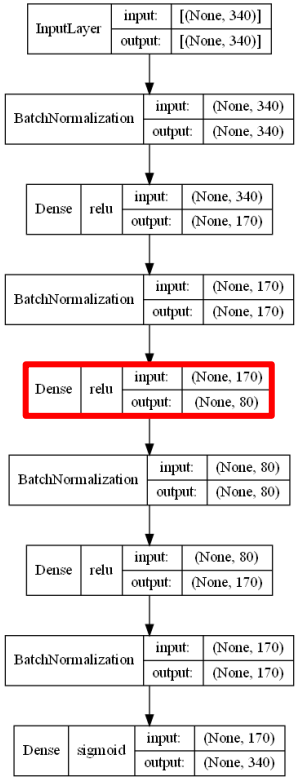
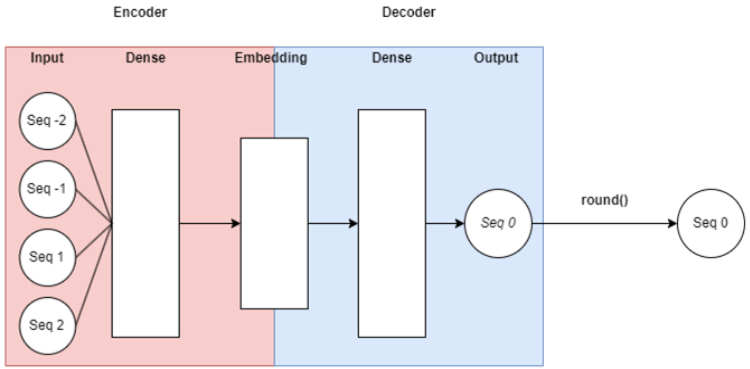


Figure 6: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	he_uniform	-
Ottimizatore	Nadam	lr=0.001, $beta\_1$=0.9, $beta\_2$=0.999
Kernel regulizer	-	-
Batch size	-	5000

Task 2 - Predittore 4 a 1 - Struttura



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

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

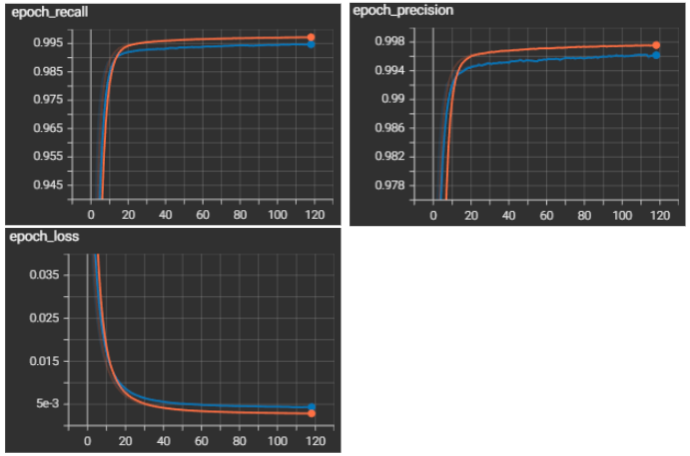
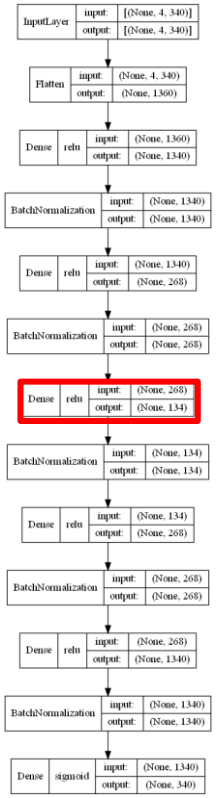


Figure 10: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	he_uniform	-
Ottimizatore	Adam	lr=$7 \cdot 10^{-5}$
Kernel regulizer	L2	$5 \cdot 10^{-6}$
Batch size	-	50

Task 2 - Predittore 4 a 1 - Modello 2

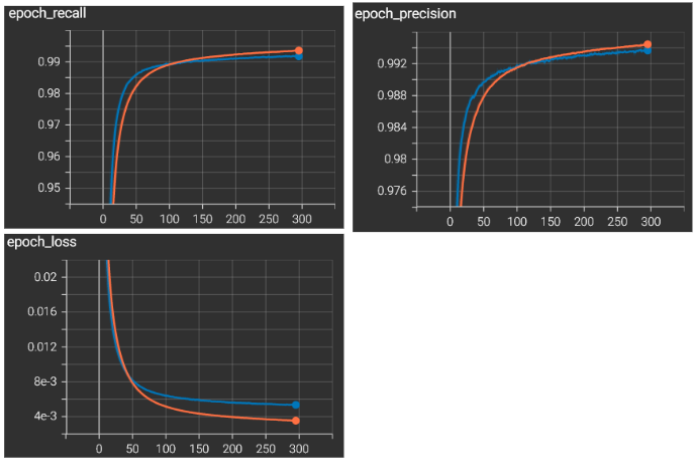
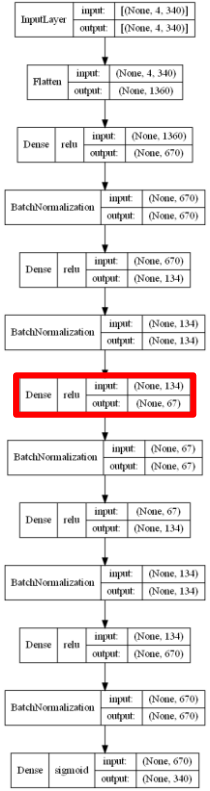
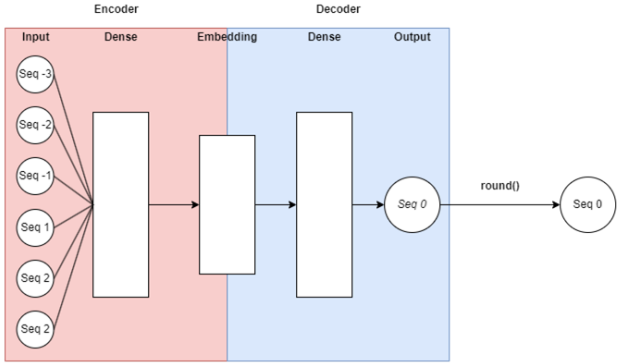


Figure 12: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	he_uniform	-
Ottimizatore	Adam	$lr{=}7 \cdot 10^{-5}$
Kernel regulizer	L2	$5 \cdot 10^{-6}$
Batch size	-	50

Task 3 - Predittore 6 a 1 - Struttura



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

Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Dimensione	Accuratezza
(340*6)2040			1020	510		1020	340	70.140%
(340*6)2040	1020	510	255	200	255		340	73.305%

Task 3 - Predittore 6 a 1 - Modello 1

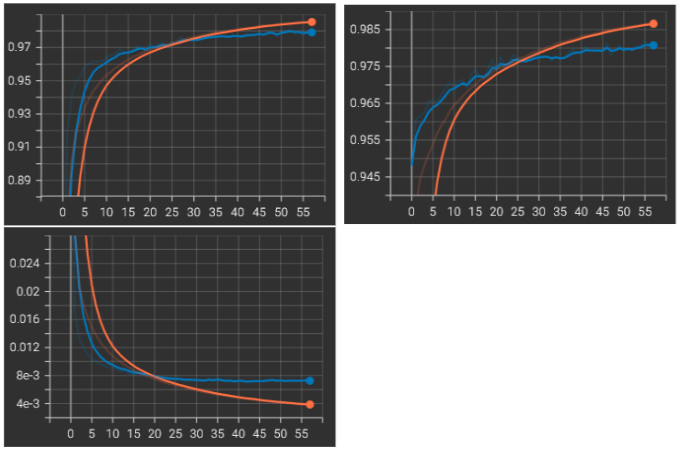
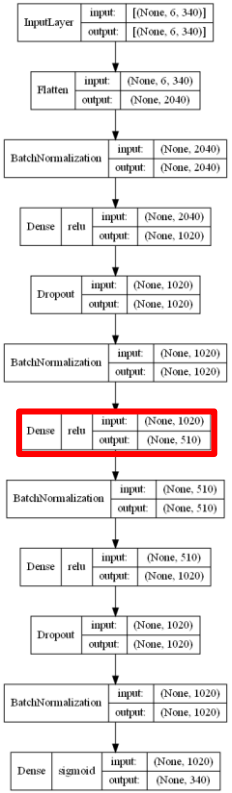


Figure 16: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



Parametri	Tipo	Valori
Kernel inizializer	he_normal	-
Optimizers	Nadam	$lr = 0.009$, $beta\_1 = 0.95$
Kernel regulizer	L2	1e-10
Batch size	-	400

Task 3 - Predittore 6 a 1 - Modello 2

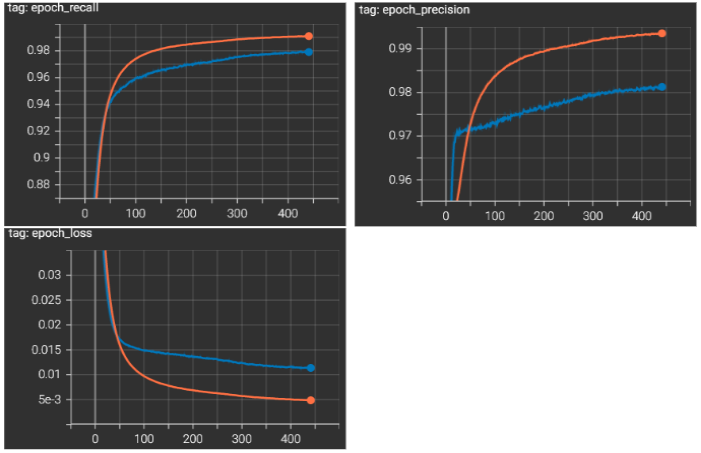
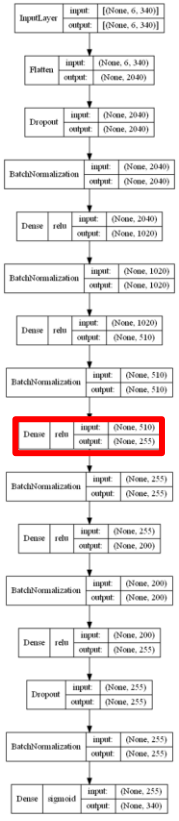


Figure 18: Andamento di recall, precision e loss su train e validation sulla base delle epoche.



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

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 overtitting e dall'elevata sparsità dei dati.