

Traning e test di una rete neurale per la ricostruzione di un'immagine sfocata v2.0

Federico Pretini, Gregorio Vecchiola, Alessandro Rongoni

Progetto didattico per il corso di Elettrotecnica a.a. 2020/2021

1 Introduzione

1.1 Sommario

Lo scopo di questo esperimento è di trovare i limiti dell'intelligenza artificiale per i quali essa riesca a ricomporre le immagini nitidamente. Questo progetto è un'espansione dei test svolti precedentemente da G. Vitali e M. La Gala, con la collaborazione del docente S. Fiori.

1.2 Premesse

Per mettere alla prova la I.A.¹ abbiamo utilizzato il tool "Sfocatura" nella sezione modifica dell'applicazione "Galleria" del modello cellulare "HONOR 7X" sistema operativo EMUI 9.1.0 versione Android 9. Indicheremo in percentuale il livello di sfocatura ricavata secondo la seguente formula:

$$x : 100 = \text{misurazione}^2 : 4\text{cm}$$

Le immagini usate sono in formato **png** e la loro risoluzione è di **240px*240px**, eccetto in due casi, Test 07_IM e Test 08_IM, di conseguenza raccomandiamo di mantenere queste condizioni.

N.B. Le immagini devono essere rigorosamente **sfocate e non mosse!**

1.3 Obiettivi

I nostri principali obiettivi erano tre:

1. Fare test su tante immagini differenti per trovare i limiti della I.A;
2. Creare un programma che permettesse di istruire il neurone con più immagini diverse automaticamente;
3. Modificare il parametro μ (Mi) nel programma_principale_manifold per ottimizzare la velocità d'apprendimento;

¹Intelligenza Artificiale

²Misurazione in cm di quanto è stata trascinata la barra della sfocatura

2 Test dei limiti

2.1 Limiti del neurone

Abbiamo elaborato **33** immagini diverse, ognuna che differiva per qualche **modifica** (diverso formato immagine, diversa risoluzione immagine, diverso livello di sfocatura, diverso zoom sul soggetto, distanza differente tra due soggetti sulla stessa foto) oppure di **soggetto** (libri, targhe, volti, automobili, scritte), per verificare cosa riuscisse a rielaborare in modo corretto e cosa, invece, non riusciva a ricostruire o, addirittura, distorcere completamente .

Questo è ciò che abbiamo notato:

- In generale, l'I.A. **NON funziona** con i volti umani, a meno che la sfocatura non sia minima;
- In generale, la distanza dal soggetto deve essere **compresa tra i 10 cm ed i 30 cm** per avere un risultato ottimale (a 40 cm di distanza non riesce a mettere a fuoco)
- La distanza dal soggetto e il livello di sfocatura sono **inversamente proporzionali**, cioè più il soggetto è distante e meno l'immagine deve essere sfocata, e viceversa;
- In generale, il livello di sfocatura dell'immagine deve essere **compresa tra l'1% e il 40%** per ottenere la maggior parte delle volte un risultato buono/ottimo. Ovviamente vale ancora quanto detto in precedenza, ovvero, in alcuni casi abbiamo ottenuto **ottimi risultati** anche su immagini sfocate al **100%** con un soggetto molto grande (lettera 'A' a 240 pixel per 240 pixel);
- Abbiamo utilizzato immagini 240px*240px per la maggior parte dei test. Abbiamo notato che comunque la risoluzione può essere **compresa tra i 200 pixel e i 300 pixel** (sia in larghezza che in altezza);
- Il formato dell'immagine **NON** influisce sul risultato finale;
- In generale, **i soggetti in primo piano risultano più nitidi rispetto ad altri in secondo piano**. Nonostante questo siamo riusciti a fare una **stima** di quanto essi potessero essere distanti gli uni dagli altri. Abbiamo visto che essi si possono distanziare per un massimo di **7 cm** con una **sfocatura massima del 30%**.

2.2 Tabella risultati

Questa tabella mostra tutte le immagini che abbiamo elaborato, ognuna con una modifica differente. Ad ogni elaborazione abbiamo assegnato un "voto" (insufficiente³, discreto⁴, buono⁵, ottimo⁶) per valutare il risultato del test.

N° TEST	IMMAGINE	MODIFICHE	VOTO
01_IM	Volto uomo	10% sfocatura	insuff.
02_IM	Volto uomo	30% sfocatura	insuff.
03_IM	Libri con sfondo	10% sfocatura, 40cm distanza	insuff.
04_IM	Libri con sfondo	20% sfocatura, 20cm distanza	ottimo
05_IM	Libri con sfondo	30% sfocatura, 20cm distanza	discreto
06_IM	Libri con sfondo	37.5% sfocatura	buono
07_IM	Libri con sfondo	37.5% sfocatura, 300px*300px	discreto
08_IM	Libri con sfondo	37.5% sfocatura, 200px*200px	discreto
09_IM	Libri con sfondo	50% sfocatura, 20cm distanza	insuff.
10_IM	Libri sfalsati di 10cm	10% sfocatura, 40cm distanza	insuff.
11_IM	Libri sfalsati di 10cm	40% sfocatura, 10cm distanza	discreto
12_IM	Libri sfalsati di 10cm	10% sfocatura, 10cm distanza	buono
13_IM	Libri sfalsati di 7cm	10% sfocatura, 10cm distanza	ottimo
14_IM	Libri sfalsati di 7cm	25% sfocatura, 10cm distanza	ottimo
15_IM	Libri sfalsati di 7cm	30% sfocatura, 10cm distanza	discreto
16_IM	Libri sfalsati di 5cm	30% sfocatura, 10cm distanza	discreto
17_IM	Libri allineati	30% sfocatura, 20cm distanza	discreto
18_IM	Libri allineati	30% sfocatura, 10cm distanza	insuff.
19_IM	Libri allineati	20% sfocatura, 10cm distanza	insuff.
20_IM	Lettera gigante	100% sfocatura	ottimo
21_IM	Targa bianca	100% sfocatura	insuff.
22_IM	Targa bianca	60% sfocatura	ottimo
23_IM	Targa bianca	70% sfocatura	buono
24_IM	Targa bianca	80% sfocatura	insuff.
25_IM	Targa arancione	60% sfocatura	buono
26_IM	Targa gialla	60% sfocatura	buono
27_IM	Targa nera	60% sfocatura	insuff.
28_IM	Targa verde-bianca	60% sfocatura	insuff.
29_IM	Targa blu	60% sfocatura	insuff.
30_IM	Targa bianca-verde	60% sfocatura	ottimo
31_IM	Targa giallo canarino	60% sfocatura	ottimo
32_IM	Targa rossa-bianca	60% sfocatura	insuff.
33_IM	Macchina con uomo	20% sfocatura	ottimo

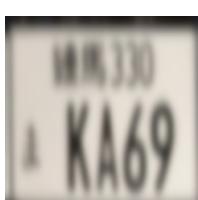
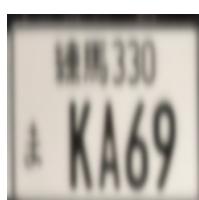
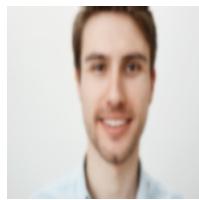
Sfondo chiaro e scritta scura aiutano molto la messa a fuoco sulle targhe.

³L'immagine non risulta nitida o addirittura viene completamente rovinata

⁴L'immagine risulta leggermente migliorata

⁵L'immagine ha un notevole miglioramento rispetto all'originale

⁶L'immagine è stata ricostruita perfettamente



2.3 Esempio di immagini

Le seguenti immagini sono degli esempi presi dalla tabella dei risultati:



(a)

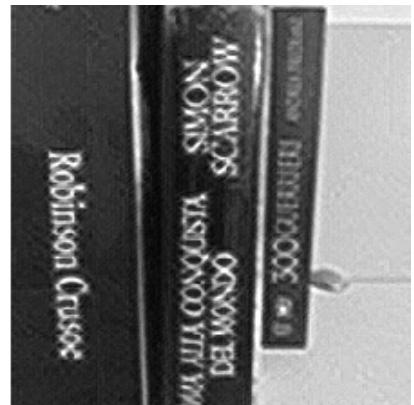


(b)

Figura 1: In questo esempio, che rappresenta il **TEST 33_IM**, è ben chiaro il concetto di rielaborazione **ottima**. Infatti la foto **(a)** è l'immagine originale (sfocata) passata al programma, mentre la foto **(b)** è ciò che esso ha rielaborato, ottenendo così un'immagine perfettamente nitida (Si riesce addirittura a vedere il profilo di chi guida l'auto!).

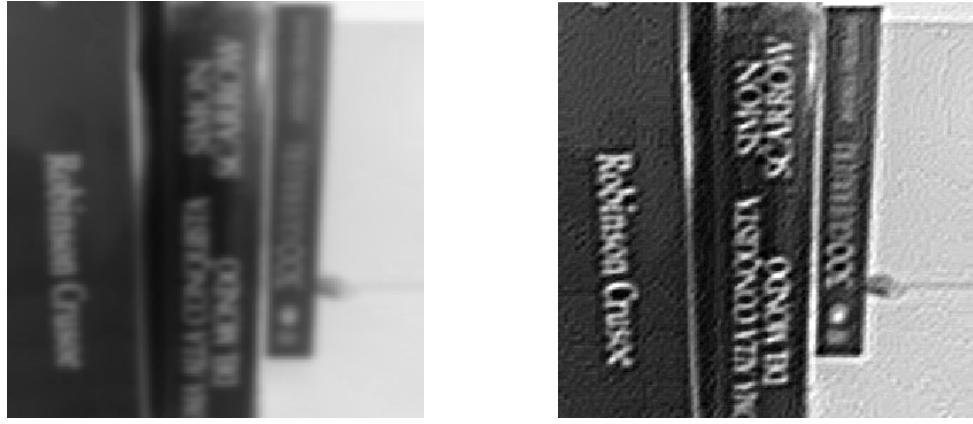


(a)



(b)

Figura 2: In queste altre due foto, **TEST 12_IM**, notiamo che l'immagine originale **(a)** è poco sfocata, e che l'intelligenza non ha avuto alcuna difficoltà a rielaborare correttamente i libri in primo piano **(b)**. Non possiamo dire lo stesso per quelli più dietro, che rimangono ancora un po' disturbati e non chiaramente nitidi. Per questo motivo gli è stato assegnato un voto **buono**.



(a)

(b)

Figura 3: In quest’altro esempio, **TEST 11_IM**, l’intelligenza ha processato la stessa immagine del **TEST 12_IM**, ma con una sfocatura maggiore. Possiamo ben notare che ha avuto difficoltà nel ricostruire non solo le scritte del libro in secondo piano, ma anche quelle dei libri poco più avanti. Per questo motivo abbiamo assegnato un voto **discreto**.



(a)

(b)

Figura 4: In quest’ultimo esempio, **TEST 02_IM** notiamo come i volti delle persone vengono completamente distorti, rendendo l’immagine quasi irriconoscibile. È inutile dire il motivo per cui abbiamo assegnato un voto **insufficiente**.

Come abbiamo detto precedentemente, lo scopo principale di questi test era quello di capire i limiti del neurone, in modo da sperimentarlo solo su un numero di campioni di immagini ristretto. Infatti, finiti questi test, abbiamo creato un programma in grado di prendere in input un numero finito di foto, con cui istruire il neurone.

3 Training & Test

3.1 Fase Training

La **FASE DI TRAINING** consiste nell’istruire l’intelligenza artificiale, facendole processare una serie di immagini differenti, con l’obbiettivo di provare ad insegnarle un metodo standard di rielaborazione. Per l’istruzione è stato selezionato un set di immagini⁷ della stessa categoria, nel nostro caso delle targhe, le quali vengono processate dal neurone, ad ogni avvio del programma, in modo casuale.

Per la creazione del programma di training abbiamo riutilizzato il programma di Umeyama, per poi adattarlo in modo che i pesi \mathbf{W} della rete neurale venissero istanziati casualmente solo al primo processo, per poi essere aggiornati in base all’immagine successiva.

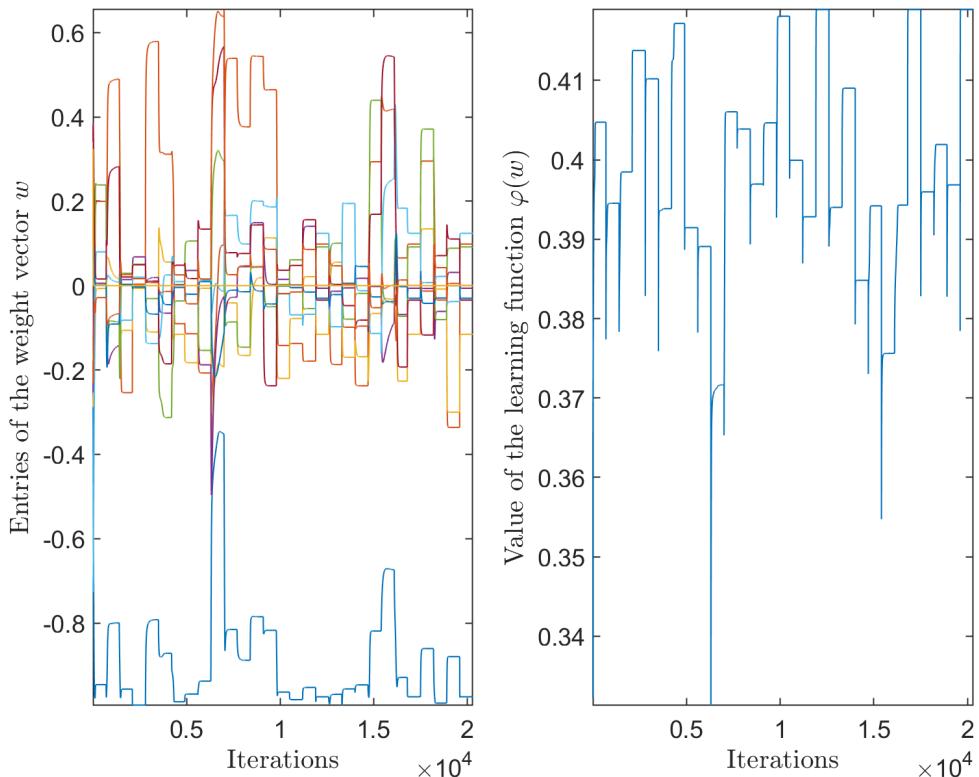
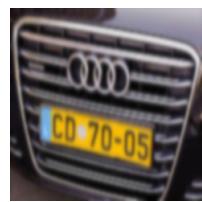


Figura 5: Esempio della convergenza della Rete Neurale.

⁷Per la precisione 29 foto.



3.2 Fase Testing

La **FASE DI TESTING** è una fase successiva a quella di Training, e consiste nell'utilizzare l'I.A. istruita per processare immagini con caratteristiche diverse. Nel nostro caso ne abbiamo selezionate due: la **prima** è un'automobile con cui l'intelligenza artificiale non è stata istruita, mentre la **seconda** rappresenta una targa facente parte del Training Set. Ci aspettiamo, quindi, che l'immagine con cui è stata istruita l'intelligenza, venga ricostruita in maniera ottimale, mentre l'altra venga ricostruita in maniera alterata.

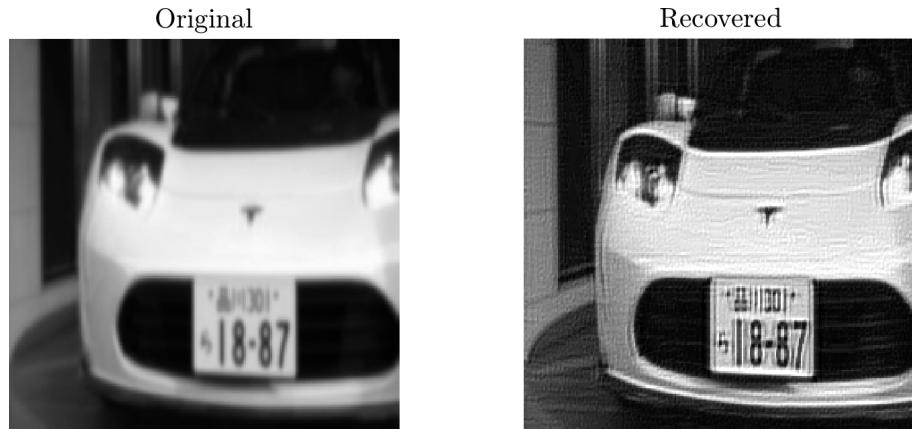


Figura 6: Immagine rielaborata **NON appartenente** al TRAINING SET

Come si può vedere, la foto non risulta perfettamente nitida, ma comunque non completamente disturbata.



Figura 7: Immagine rielaborata **appartenente** al TRAINING SET

Purtroppo, anche quest'ultima immagine non risulta completamente nitida.

3.3 Osservazioni

Come abbiamo visto nella sottosezione precedente, i risultati ottenuti non hanno rispecchiato le nostre aspettative. Il risultato, però, non sempre è il medesimo, infatti, svolgendo ulteriori avvii di training, si nota come il risultato **cambi al variare dell'ordine** in cui le immagini sono state processate nella fase di istruzione. In base a ciò, avremmo in maniera casuale, un risultato soddisfacente o non.

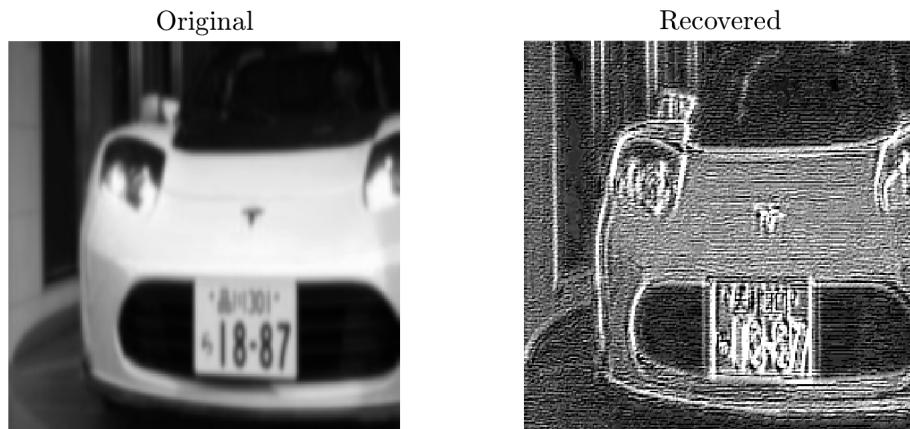


Figura 8: Un ulteriore esempio della rielaborazione dell'immagine **NON appartenente** al Training Set, ma con una disposizione differente delle foto processate per l'apprendimento.

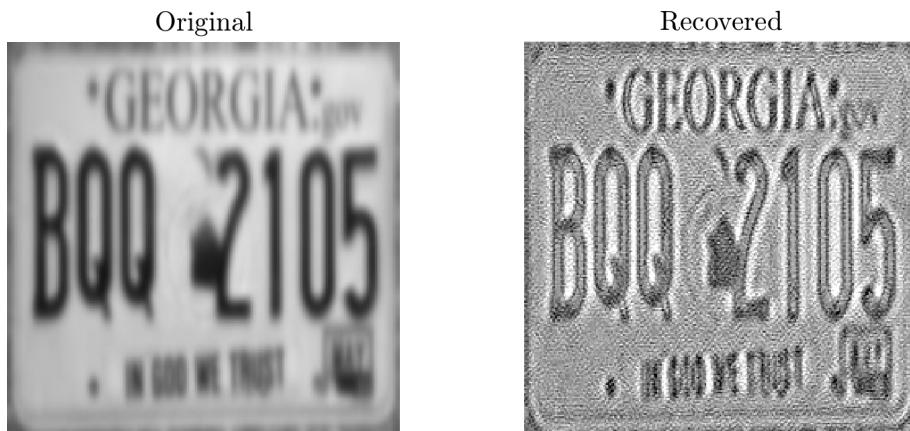


Figura 9: Rielaborazione dell'immagine **appartenente** al Training Set, con differente ordine di foto processate nella fase di apprendimento.

In questo caso, entrambe le immagini vengono completamente deteriorate. In altri, paradossalmente, l'immagine che non aveva mai processato prima, può risultare molto più nitida di quella "studiata" in precedenza durante la fase di Training. Nella pagina seguente vedremo ulteriori esempi.

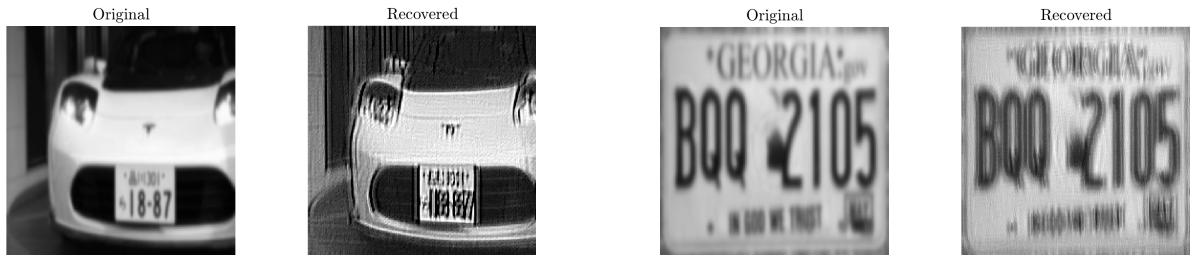


Figura 10: Esempio 1



Figura 11: Esempio 2



Figura 12: Esempio 3

Possiamo quindi concludere che **non è possibile insegnare al neurone un metodo standard** per rielaborare correttamente le immagini.

Nel prossimo capitolo lavoreremo con un diverso programma MATLAB, il **Manifold**, il quale cercheremo di ottimizzarlo modificando il parametro di apprendimento e lo compareremo con quello di Umeyama.

4 Ottimizzazione Manifold

4.1 Modifica parametro μ

Per ottimizzare il programma Manifold, siamo andati a modificare il **learning parameter** μ (dichiarato nel programma come M_i). Impostare un valore corretto di quest'ultimo è molto importante, in quanto permette la convergenza veloce e stabile nei metodi di ottimizzazione non lineari. Attraverso dei test, però, ci siamo resi conto che questo valore ha un **limite**, che può variare a seconda dell'immagine. Superata la soglia di questo, i pesi w e la *rete neurale* iniziano ad oscillare, ed a quel punto il risultato dipende dall'ultimo valore registrato nell'oscillazione.

In sostanza, visto che la **matrice W** viene inizialmente istanziata in modo randomico unito al fatto che la **funzione di apprendimento** non converge, fanno in modo che la rielaborazione dell'immagine sia totalmente casuale, a seconda, appunto, dell'ultimo valore della curva, che può essere un **PUNTO DI MASSIMO** (immagine rielaborata in maniera ottimale) oppure un **PUNTO DI MINIMO** (immagine rielaborata in maniera pessima).

Per dimostrare ciò abbiamo utilizzato la Figura 13 come immagine test, e i grafici di seguito, rappresentano l'andamento delle curve in base ai cambiamenti del μ .



Figura 13: Immagine usata per i grafici, la si può ritrovare nella cartella Targhe_Test/Test_2

4.2 Risultati della convergenza con diverso μ

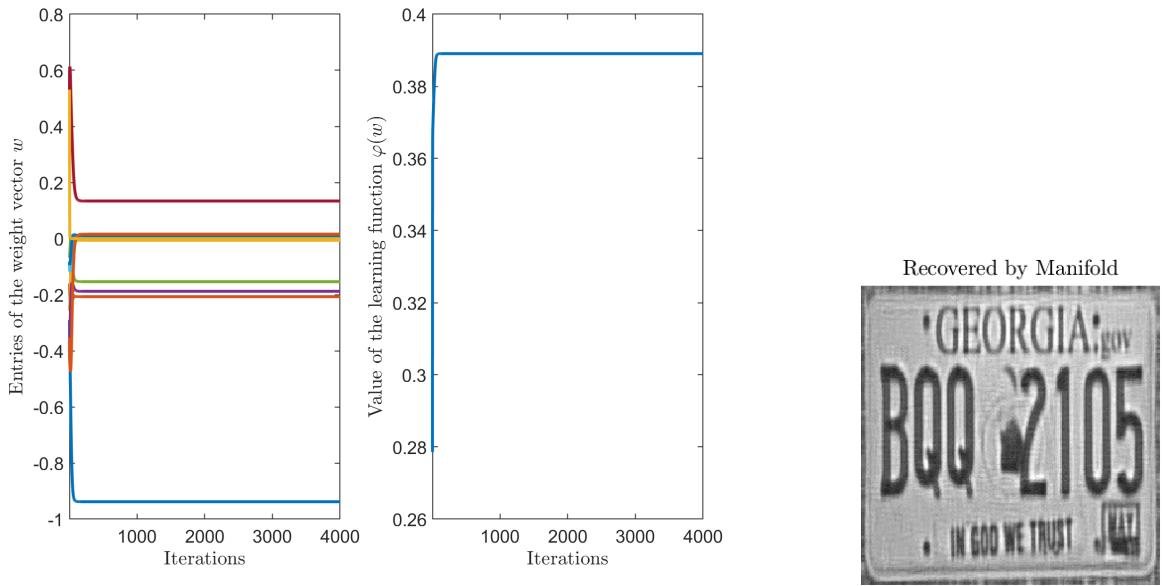


Figura 14: Convergenza con $\mu=0.0000154$ con rispettiva ricostruzione

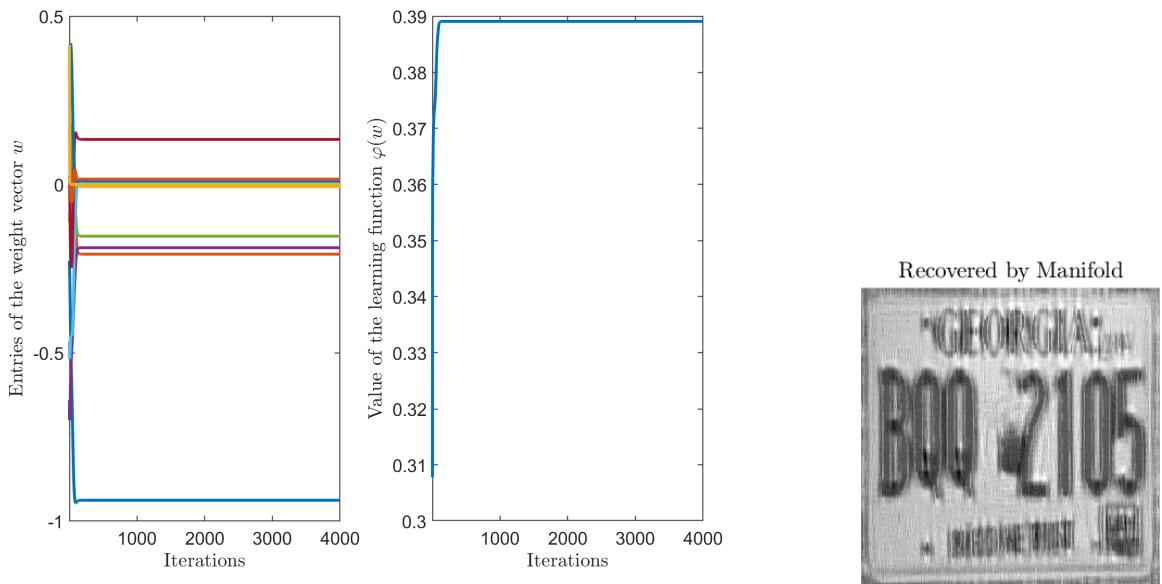


Figura 15: Convergenza con $\mu=0.00002$ con rispettiva ricostruzione

In questi due esempi le curve raggiungono un picco massimo per poi **stabilizzarsi** e rimanere **costanti** (convergono).

Di seguito vedremo tutti esempi in cui le curve oscillano.

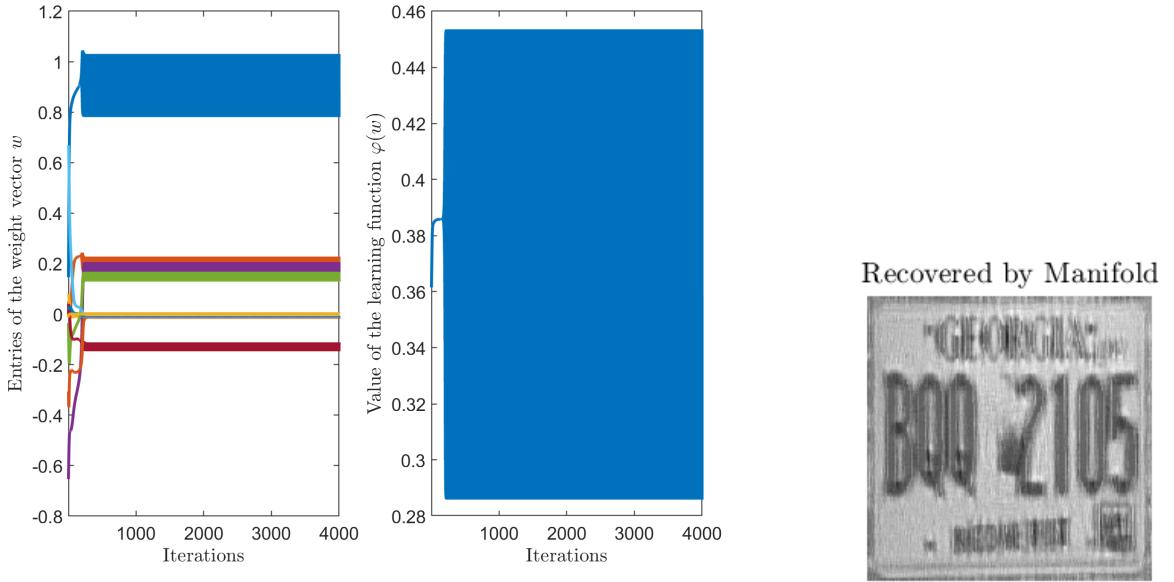


Figura 16: Oscillazione con $\mu=0.00003$ con rispettiva ricostruzione con punto di minimo

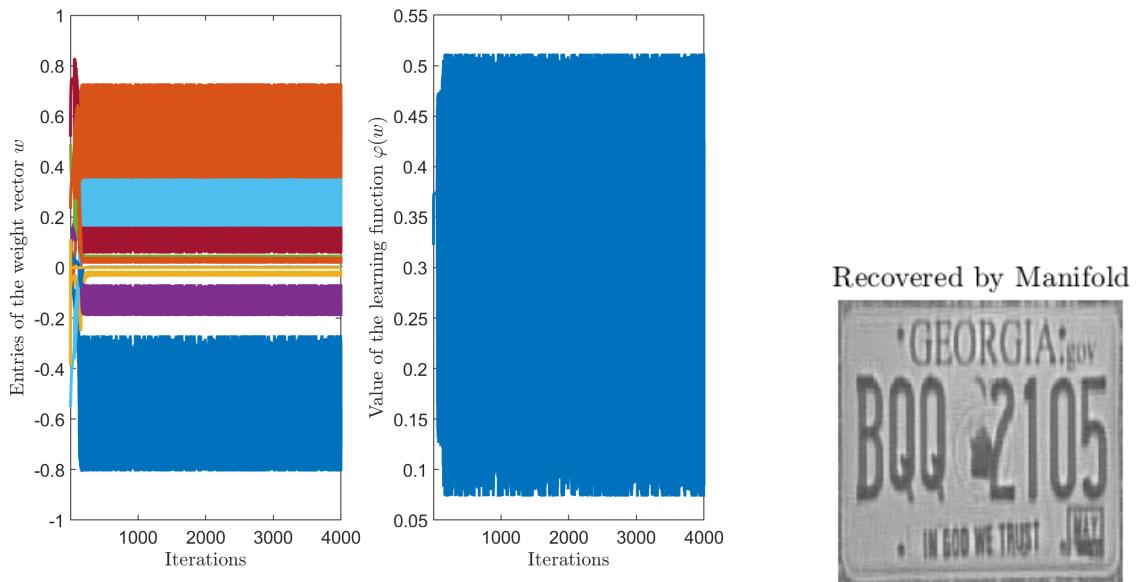


Figura 17: Oscillazione con $\mu=0.00004$ con rispettiva ricostruzione con punto di massimo

In generale, una volta superato il limite, all'aumentare di μ , aumenta anche l'ampiezza delle oscillazioni, raggiungendo, così, punti di massimo molto più alti del normale.

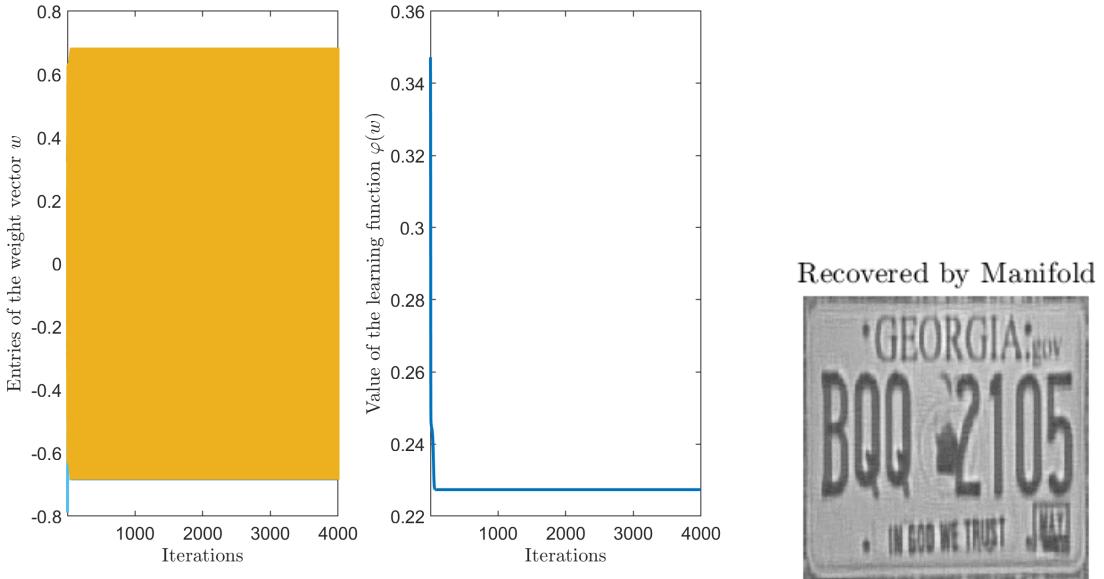


Figura 18: Convergenza con $\mu=0.00007$ con rispettiva ricostruzione

Quest’ultimo caso è molto interessante, in quanto, probabilmente, i pesi oscillano tra due configurazioni a cui corrisponde la stessa φ . Ad esempio, se w oscilla tra -1 e +1 e $\varphi = w^2$, allora $\varphi(1) = \varphi(-1) = 1$. Quindi φ sembra **costante**, ma w **varia**.

4.3 Comparazione Umeyama e Manifold con stesso μ

Attraverso il **programma_principale_comparison**, abbiamo ottenuto dei grafici comparativi tra le funzioni di apprendimento di Umeyama e Manifold per una stessa immagine, con un valore di μ identico. Abbiamo osservato che:

- Con un valore di μ intorno a **0.00001**, la curva di apprendimento della rete neurale del programma Manifold converge più velocemente rispetto a quello di Umeyama (Figura 19);
- Al **diminuire** di μ , le due curve convergono pressoché in modo identico (Figura 20);

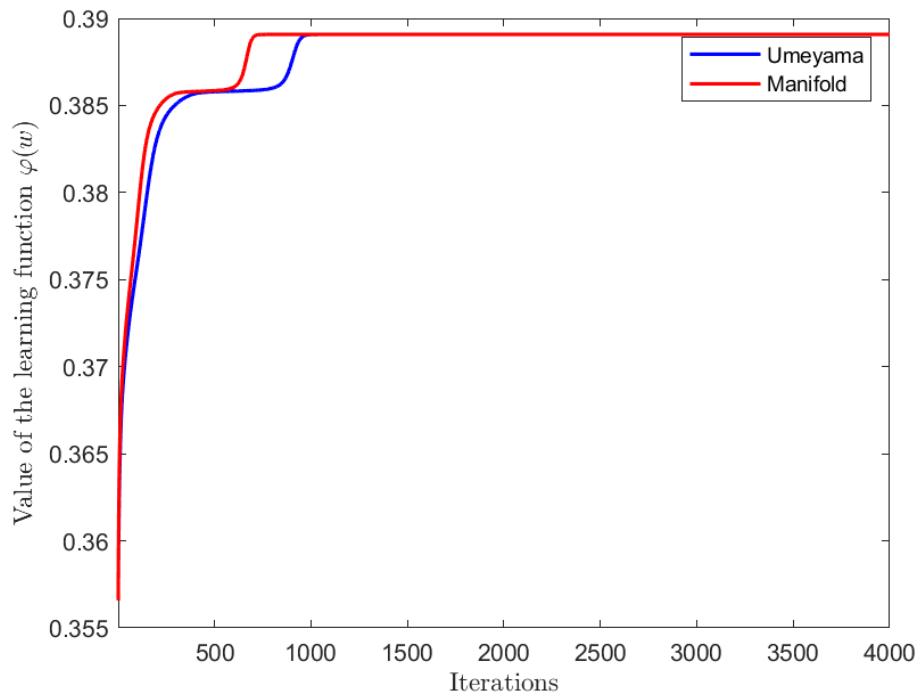


Figura 19: Convergenze delle reti neurali di Umeyama e Manifold con $\mu=0.00001$

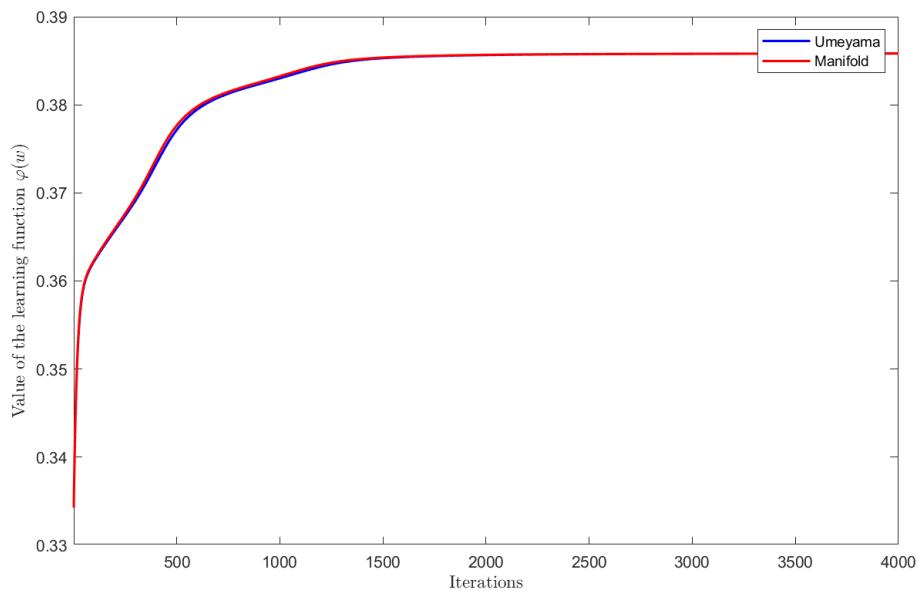


Figura 20: Convergenze delle reti neurali di Umeyama e Manifold con $\mu=0.000001$

5 Finale

5.1 Conclusioni

In questo progetto abbiamo utilizzato i programmi forniti dal professore, di Umeyama e Manifold, e abbiamo effettuato molti test nel tentativo di conseguire gli obiettivi sopra citati che ci eravamo prefissati, e che sono stati, poi, raggiunti correttamente. Durante i nostri test, però, abbiamo notato le seguenti cose:

- Nel caso del secondo obiettivo siamo riusciti a creare un programma in grado di istruire l'intelligenza artificiale, ma il neurone non ha risposto come ci immaginavamo. Infatti, ci siamo resi conto che era in grado di processare una sola immagine alla volta, senza tenere in considerazione quello che aveva imparato prima, pur mantenendo i pesi w di ciò che aveva processato prima.
- Per quanto riguarda il terzo ed ultimo obiettivo, abbiamo notato che ogni immagine ha un limite diverso di μ , che, superato, farà oscillare le curve dei pesi w e dell'apprendimento φ .

Nella prossima sottosezione proponiamo un possibile utilizzo di queste oscillazioni per ottenere immagini perfettamente nitide con un valore di μ alto.

5.2 Lavoro futuro

Prendendo spunto dalle osservazioni fatte da noi riguardo al terzo obiettivo, i futuri studenti potrebbero pensare di utilizzare queste oscillazioni per ottenere dei picchi di apprendimento molto più alti rispetto alla solita convergenza della rete neurale. Per fare ciò basterebbe ricercare un punto di massimo dell'oscillazione (all'interno della matrice "conv"). Una volta trovata la posizione al suo interno, si utilizza il suo posizionamento per andare rintracciare all'interno della matrice W(nella medesima posizione) i pesi w migliori per la ricostruzione dell'immagine. In questo caso la ricostruzione potrebbe essere addirittura migliore del caso in cui l'apprendimento converge semplicemente.

5.3 Riferimenti bibliografici

1. Shinji Umeyama, Blind Deconvolution of Images using Gabor Filters and Independent Component Analysis, ICA2003, Nara, Japan.
2. Shinji Umeyama, Blind deconvolution of Blurred Images by Use of ICA, Electronics and Communications in Japan, Part 3, Vol.84, No. 12,2001
3. RICOSTRUZIONE DI UN'IMMAGINE SFOCATA UTILIZZANDO I FILTRI DI GABOR E L'ALGORITMO INDEPENDENT COMPONENT ANALYSIS GIACOMO VITALI E MARCO LA GALA Progetto didattico per il corso di Analisi Numerica a.a. 2017/2018 - Prof. Simone Fiori