

Challenge 3 ML

Matteo Vicenzino

May 2025

Si vuole analizzare l'effetto delle Reti Neurali Convoluzionali e Fully Connected sul dataset Kuzushiji-MINST, che contiene 70.000 immagini di 28×28 pixel rappresentanti caratteri giapponesi, divise in 10 classi corrispondenti a diversi simboli hiragana. Si studierà l'effetto della regolarizzazione tramite Dropout e di diversi ottimizzatori sulle prestazioni delle reti neurali durante l'addestramento.

Data Exploration

Stampando alcune informazioni utili sulle foto si nota che esse presentano un solo canale colore, sono divise in 60.000 dati di train e 10.000 in un dataset di test.

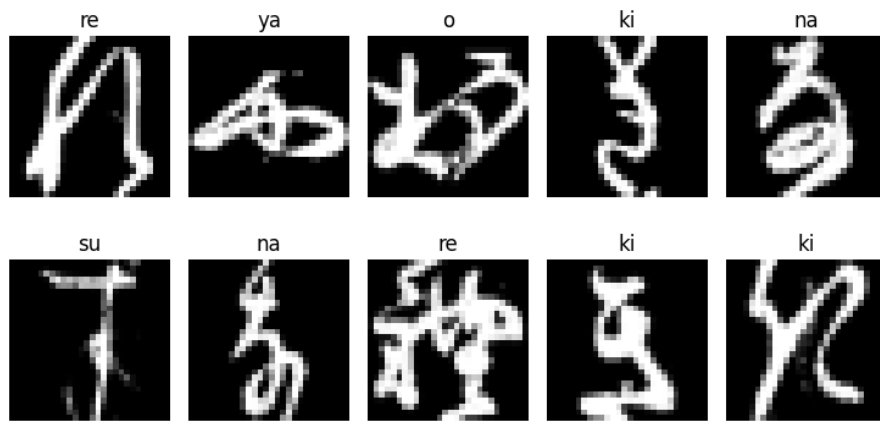


Figure 1: Prime 10 immagini del dataset di train

Inoltre si è diviso il dataset di train attraverso la funzione `random_split` di pytorch nel seguente modo:

- Train set 80% (48.000 immagini)
- Validation set 20% (12.000 immagini)

Alle immagini inoltre è stata applicata una rotazione random di 10 gradi in modo di aumentare la variabilità del dataset e migliorare la capacità di generalizzazione del modello, rendendo il sistema più robusto a piccole variazioni nell'orientamento dei caratteri.

Le immagini vengono appiattite in vettori di 784 valori, ciascuno rappresentante l'intensità normalizzata di un pixel. Questi vettori alimentano layer di input della rete neurale, dove ogni neurone riceve un singolo valore di pixel, trasmettendo l'informazione ai layers successivi. La funzione di attivazione usata è stata la ReLU. Il numero di layer di output invece è impostato a 10 in modo che ogni immagine venga mappata in una delle 10 classi del dataset.

Fully Connected NN

Si inizia sviluppando alcuni modelli di rete neurale con layer fully connected, testando diverse configurazioni di hidden layers e diversi iperparametri. La dimensione della Batch è fissa a 32 e il numero di epoch è pari a 10. Di seguito sono riportati i diversi modelli allenati.

Model	Hidden Layers	Hidden Layers Size	Learning Rate	Optimiser	Dropout Rate	Train Acc.	Val. Acc.	Test Acc.
FCNN	2	256, 256	0.005	SGD	0.3	92.60%	94.06%	88.31%
FCNN	3	512, 256, 64	0.01	SGD	0.2	94.97%	94.84%	89.46%
FCNN	3	512, 256, 256	0.0005	Adam	0.4	92.76%	94.82%	88.81%

Table 1: Risultati dei modelli FCNN con diversi iperparametri e configurazioni

La rete con tre layer nascosti (512, 256, 64 neuroni) e ottimizzatore SGD ha ottenuto le migliori prestazioni, raggiungendo un’accuratezza del 89.46% sul test set e del 94.97% sul training set. Questa configurazione, caratterizzata da un learning rate di 0.01 e un dropout rate di 0.2, ha superato la rete a due layer (256, 256) con SGD, che ha registrato un’accuratezza dell’88.31% sul test set. La rete con tre layer (512, 256, 256) e ottimizzatore Adam ha mostrato prestazioni leggermente inferiori, con un’accuratezza dell’88.81% sul test set, nonostante un dropout rate più elevato (0.4) e un learning rate molto ridotto (0.0005). I risultati suggeriscono che l’architettura con numero di neuroni decrescente nei layer nascosti, abbinata all’ottimizzatore SGD, offre il miglior compromesso tra capacità di generalizzazione e accuratezza per questo specifico task di classificazione.

Convolutional NN

Dopo lo sviluppo di reti neurali fully connected con diverse configurazioni, si passa all’implementazione di reti convoluzionali (CNN), più efficaci per il riconoscimento di immagini. La CNN utilizzata è composta da tre blocchi principali, ciascuno formato da un layer convoluzionale con kernel 3×3 , utile per catturare caratteristiche locali, seguito da una funzione di attivazione ReLU che introduce non linearità, e da un’operazione di MaxPooling (2×2), che riduce la dimensionalità spaziale e contribuisce a rendere il modello più robusto a piccole traslazioni dell’immagine (proprietà dell’invarianza). La rete si completa con la Batch Normalization, che aiuta a stabilizzare e velocizzare l’apprendimento, e due layer fully connected finali per la classificazione, supportati da Dropout per ridurre l’overfitting. Di seguito sono riportati i diversi modelli allenati.

Model	Conv Layers	Conv Layers Size	Learning Rate	Optimiser	Dropout Rate	Train Acc.	Val. Acc.	Test Acc.
CNN	2	32, 64	0.005	Adam	0.5	94.13%	97.58%	93.77%
CNN	2	64, 128	0.01	SGD	0.5	95.86%	97.72%	94.56%
CNN	3	64, 128, 256	0.005	SGD	0.5	98.65%	98.50%	95.41%
CNN	3	32, 64, 128	0.001	Adam	0.5	98.42%	98.63%	96.03%
CNN	3	16, 32, 64	0.005	Adam	0.4	97.75%	98.48%	95.75%

Table 2: Risultati dei modelli CNN con diverse configurazioni di layer convoluzionali, tassi di dropout e ottimizzatori

La configurazione con le migliori prestazioni è la rete CNN a tre layer convoluzionali di dimensioni 32, 64, 128, ottimizzata con Adam, che ha raggiunto un’accuratezza del 96.03% sul test set, impiegando un learning rate di 0.001 e un dropout del 0.5. Al secondo posto si colloca la rete a tre layer (16, 32, 64) sempre con Adam (learning rate 0.005, dropout 0.4), con un’accuratezza del 95.75%. Subito dopo troviamo la configurazione a tre layer (64, 128, 256) con SGD (learning rate 0.005, dropout 0.5), che ha ottenuto il 95.41% sul test set. Le reti a due layer convoluzionali—(32, 64) con Adam e (64, 128) con SGD—hanno invece raggiunto il 93.77% e il 94.56% rispettivamente. Questi risultati confermano come un’architettura più profonda e un tasso di dropout adeguato possano migliorare significativamente la capacità di generalizzazione.

Results

In sintesi, le reti convoluzionali (CNN) hanno dimostrato prestazioni complessivamente superiori rispetto ai modelli fully connected (FCNN), raggiungendo accuratze più elevate sul test

set grazie alla capacità di estrarre in modo più efficace le caratteristiche spaziali dei dati.

Inoltre la regolarizzazione Dropout ha una funzione fondamentale nel limitare l'overfitting, infatti dalle figure 2 (dorpout minore) e 3 (dropout maggiore) si nota che il modello con Dropout rate minore (0.4) presenta un leggero overfitting in quanto l'accuratezza sul dataset di validation si stabilizza, mentre quella sul trainig set continua ad aumentare in maniera costante. Questo è indice che il modello ha memorizzato le immagini del dtaset di train e non riesce a generalizzare bene sui dataset di validation e test

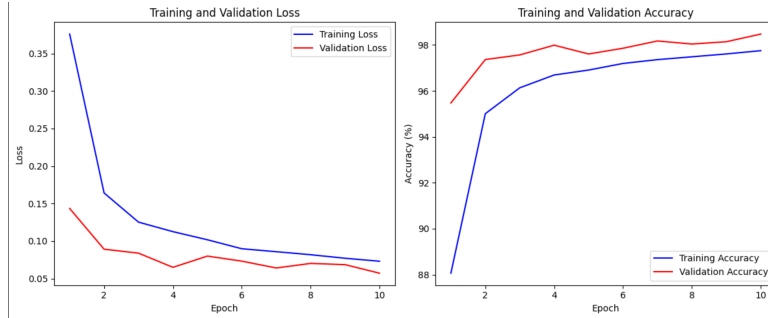


Figure 2: Loss e Accuracy dell'ultimo modello CNN, train e validation

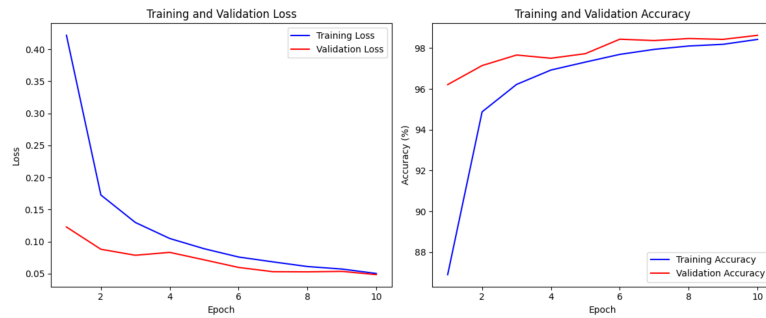


Figure 3: Loss e Accuracy del penultimo modello CNN, train e validation

Oltre all'accuratezza, è utile analizzare la confusion matrix per comprendere quali classi il modello tende a confondere maggiormente, evidenziando eventuali squilibri o ambiguità nei dati. Differenze tra le prestazioni su training, validation e test set possono indicare overfitting o una scarsa generalizzazione. Sebbene l'accuracy fornisca una misura sintetica delle prestazioni, potrebbe non essere sufficiente in presenza di classi sbilanciate: in questi casi, metriche come precision, recall o F1-score risultano più informative.

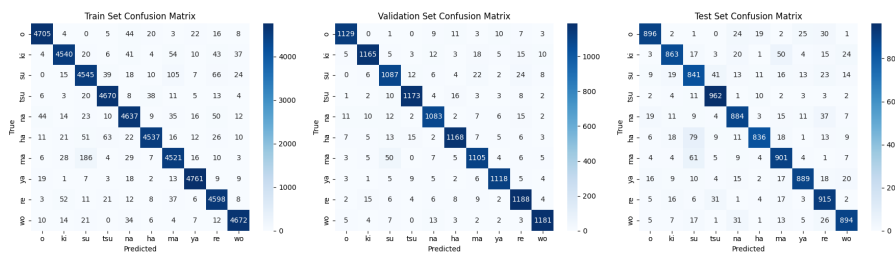


Figure 4: Confusion matrix del penultimo modello