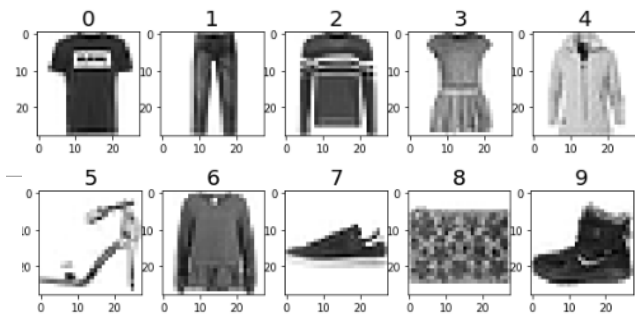


Gruppo: Baccoli, Guttilla, Trovato, Turco

Fashion MNist

Riconoscimento e classificazione di immagini di articoli di Zalando.

Il dataset di riferimento, Fashion Mnist, consiste in un'evoluzione del classico MNist, presentando 60000 immagini di articoli d'abbigliamento nel set di training e 10000 immagini nel test, nel formato 28x28 e suddivise in 10 classi: t-shirt/top, pantaloni, maglioni, vestiti, giacche, sandali, camicie, sneaker, borse e scarponcini.

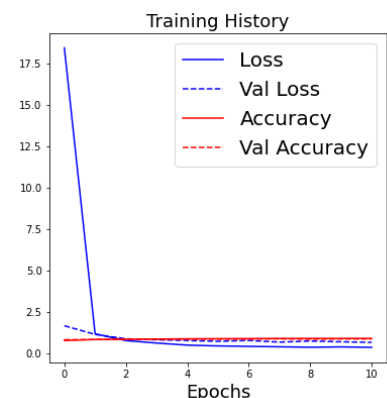


Da una prima analisi del dataset, distribuzione perfetta dei dati nelle 10 classi, ognuna delle quali contiene 6000 immagini per il training e 1000 per il test. Non essendoci dati mancanti o altre operazioni di data cleaning da completare, abbiamo proceduto con la divisione del training set

in train effettivo e validation set con una proporzione di 80 /20, mantenendo la stratificazione sulle label.

Baseline model

Come primo tentativo abbiamo costruito un 'baseline model', che abbiamo cercato di mantenere il più semplice possibile. Abbiamo trasformato le immagini da dare in input al modello come unico vettore flat e le label in vettori binari con un'operazione di one-hot encoding. Dopo aver costruito il modello con tre layer ed effettuato il fit, la valutazione del modello ci ha mostrato una percentuale di errore del 14.76 e un'accuracy del modello sul test set del 0.84.

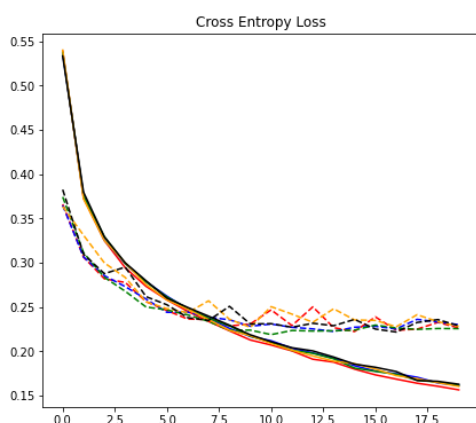
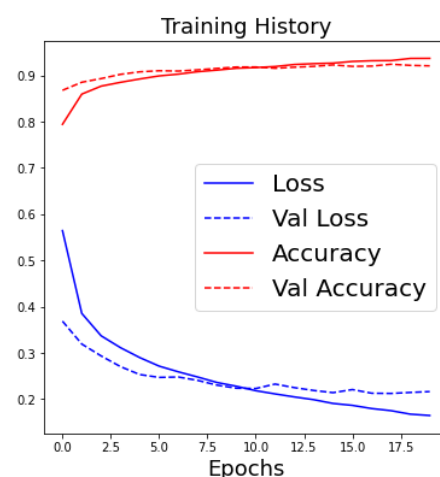


Convolutional Networks

Visto che il problema di classificazione su immagini viene affrontato con prestazioni migliori da reti convoluzionali, abbiamo deciso di costruire diversi modelli di CNN. Per poter utilizzare le reti convoluzionali abbiamo trasformato i dati, prima linearizzando le immagini con un reshape che aggiungesse un'ulteriore dimensione all'immagine, quella del channel del colore (in questo caso solo uno per il grigio), e successivamente normalizzando per il valore massimo che può assumere un pixel. Per quanto riguarda le label, invece, abbiamo nuovamente effettuato un'operazione di one-hot encoding, trasformando le label in vettori binari.

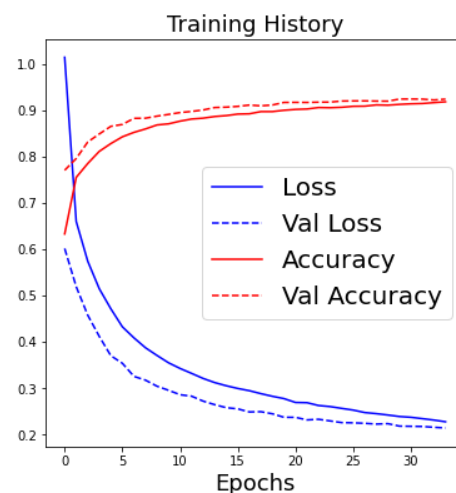
- **Cross-validation** Nel primo modello di CNN che abbiamo provato a costruire, abbiamo deciso di costruire un modello con una struttura semplice a cui poi applicare la Cross-validation. Abbiamo limitato il numero di layer Conv2D a 2, con un numero di filtri non troppo grande (32 e 64). Come funzione di attivazione abbiamo scelto la relu per tutti i livelli, tranne l'ultimo in cui abbiamo utilizzato la softmax, per ottenere la probabilità finale di appartenenza alle classi.

In prima battuta, abbiamo eseguito il fitting del modello su 20 epoche e ne abbiamo valutato la bontà utilizzando i due set di training e validation, per cui abbiamo calcolato la percentuale di errore e, in particolare, visualizzato il plot della training history. Grazie al plot, infatti, ci è stato possibile osservare che dalla decima epoca in poi il modello entra in overfitting; abbiamo dunque deciso di rinizializzare il modello e fittarlo su 10 epoche. Con questo modello, siamo riusciti ad ottenere 0.90 di accuracy e, per cercare di migliorarlo, abbiamo deciso di applicare una cross validation a 5 fold sul training set

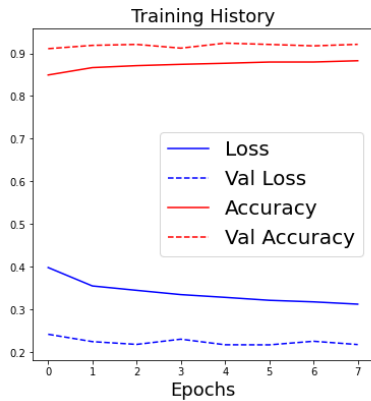


originale, prima dello split in train e validation, per poter utilizzare tutte le immagini del train. Per valutare il risultato della K-Fold, abbiamo utilizzato un plot per la loss e l'accuracy di tutti e cinque tentativi. Anche in questo caso, grazie al plot abbiamo notato l'inizio dell'overfitting intorno alla settima epoca, dunque, nel momento del fit del modello abbiamo ridotto ulteriormente le epoche a 7. Il modello così fittato, ha migliorato leggermente l'accuracy sul test, arrivando a 0.912.

- Hyperparameter Tuning** Un ulteriore tentativo che abbiamo provato con le reti convoluzionali è stato la ricerca dei parametri migliori per il modello grazie al tuning automatico degli iperparametri sfruttando il Keras-Tuner della libreria keras. Abbiamo mantenuto un modello non troppo complesso, con diversi layer ma relativamente pochi filtri e unità nel livello denso, e abbiamo stabilito diversi range di valori per cui ottimizzare gli iperparametri nei diversi layer della rete, quali: il numero di filtri, il valore di dropout, il numero di unità del livello denso e diversi tipi di funzioni di attivazione per quest'ultimo livello. Dopo la random search, abbiamo individuato gli iperparametri migliori per il modello e abbiamo effettuato il fit. La predizione sul test set e la valutazione dell'accuracy ci ha portato ad un risultato dello 0.917, piuttosto in linea con i risultati del modello precedente.

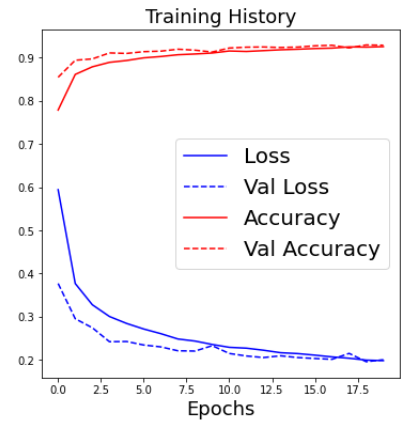


- **Data Augmentation** Nel terzo modello con reti convoluzionali che abbiamo provato, abbiamo pensato di applicare la data augmentation ai dati, in modo da aumentare il numero di immagini su cui effettuare l'allenamento del modello. Il costo computazionale inferiore rispetto a tuner automatico e



Training history del modello con data augmentation

cross-validation, ci ha permesso di costruire una struttura di rete più complessa, con tre layer di Conv2D e un numero crescente di filtri e con due livelli densi di 256 e 128 unità rispettivamente. Il tentativo, però, è stato piuttosto fallimentare, in quanto la metrica dell'accuracy calcolata sulla predizione nel test set senza l'aumento dei dati si è rivelata leggermente più alta di quella del modello fittato sui dati aumentati, con, nel primo caso, un risultato dello 0.919 e, nel secondo, dello 0.910.



Training history del modello senza data augmentation

Conclusioni

In conclusione, nonostante i diversi modelli che abbiamo provato a strutturare e le diverse tecniche che abbiamo applicato, non è stato possibile riscontare un cambiamento significativo dell'accuracy sul test set nei diversi modelli, in quanto tutti si attestano fra lo 0.91 e lo 0.92. Ugualmente, un'altra particolarità che accomuna tutti i modelli è la difficoltà nel classificare correttamente le classi 0 (t-shirt/top), 2 (maglioni), 4 (giacche) e, soprattutto, 6 (camicie), per cui i valori sia di precision che di recall risultano aggirarsi sempre intorno allo 0.8 e, nel caso della sesta classe, scendono fino allo 0.70, evidenziando le difficoltà dei modelli nel distinguere la stessa tipologia di indumento.