Keras

1) Image classification

MNIST - pre processed data

Downloading datasets

(x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data()

Pre process

1) Image normalization
2) Change in dimensionality

Model

1) Convo 2D
- extracts features and positions

keras.layers.Conv2D(64, Kernal_size(3,3), activation="relu")

64 → number of filters
(3,3) → 3×3 matrix
"relu" → gives 0 is -ve value

keras.layers.Dropout(0.5)

2) ~~Dense~~ keras.layers.Dense(num_classes, activation="Softmax")

"Softmax" → Highest probability feature

Dropout
Reduces the neuron relation for better accuracy

pooling

reduces the matrix makes a new image similer to og

~~Dense~~
model.summary() → gives parameters

Compiling model

```
model.compile(
loss = keras.SparseCategoricalCrossentropy(),
                ↑ used when
        # binary in^two ~~dess~~ class
          eg (male, female)
optimizer = keras.optimizer.
    ↑       Adam(learning_rate = 1e-3),
    2
# variance in ~~value~~ ~~evil~~ predicted value
  vs actual value
Adam = Root mean squared +
       stochastic gradient descent
metrics = [ keras.metrics.
  SparseCategoricalAccuracy(name = "acc"),
                ↑
  ],
) # gives accuracy of model
```

# callback

↗

if a criteria is met training stops

epoch → iteration

```
model.fit( x_train, y_train,
    batch_size = batch_size,
# can't use 60,000 data at once
    epochs = epochs, validation_split=0.15,
    callbacks = callbacks)
```