Report 1

109030605

黃昭學

Data preprocessing

An image with 28×28 pixels $\begin{bmatrix} 0 & \cdots & 123 \\ \vdots & \ddots & \vdots \\ 95 & \cdots & 0 \end{bmatrix}$

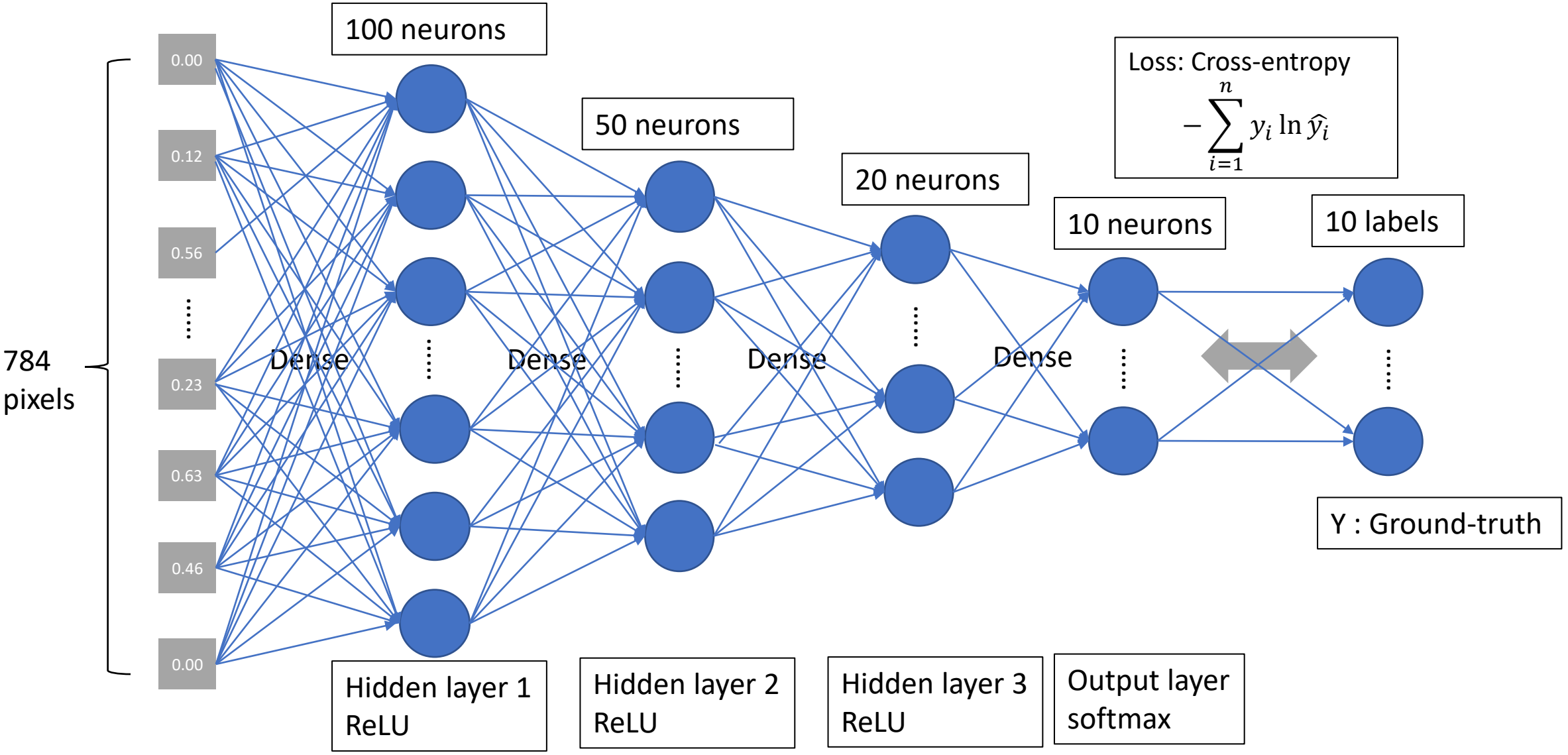
The maximum value of image pixels is 255

Normalization
(min-max scaler)

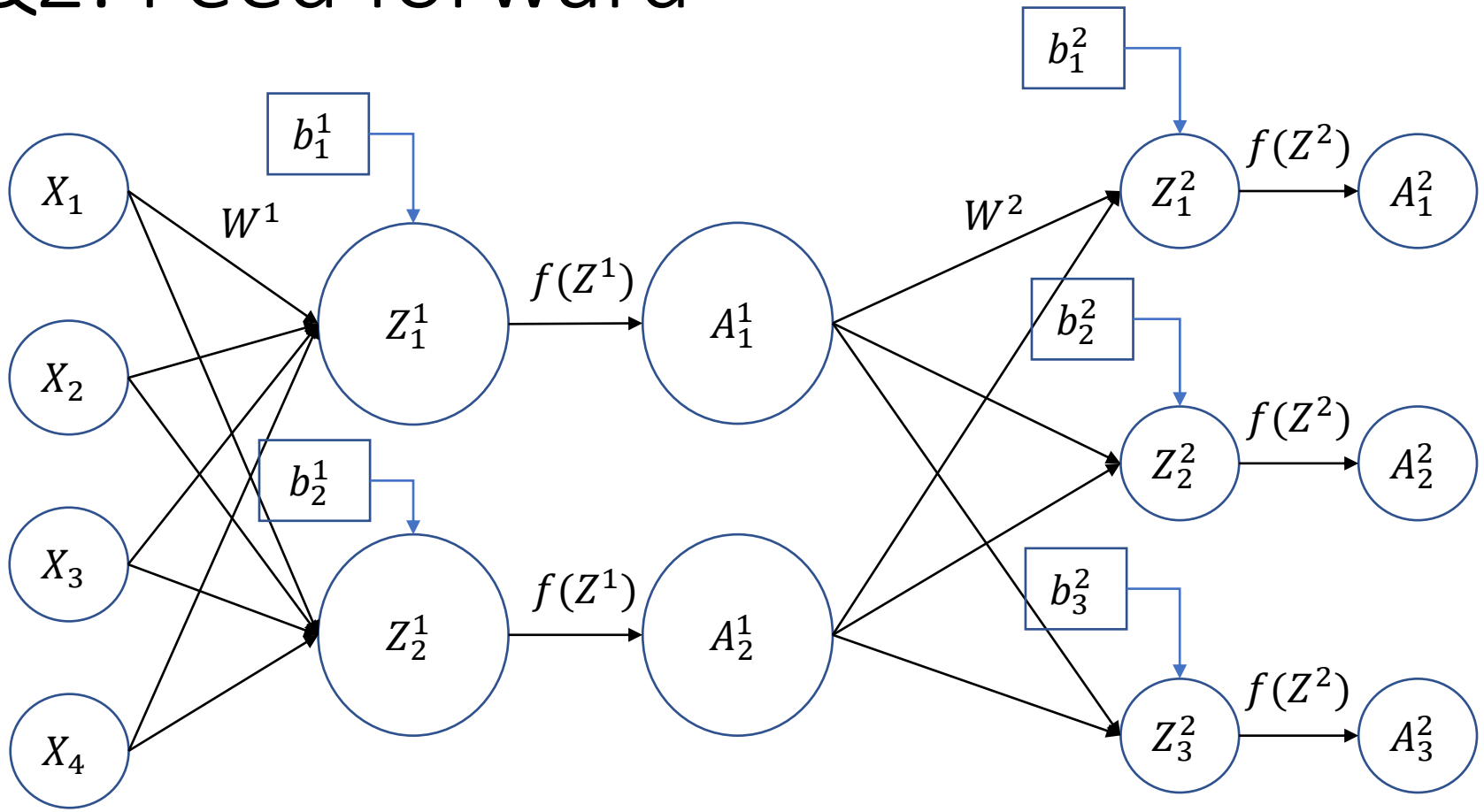
An image with 28×28 pixels $\begin{bmatrix} 0 & \cdots & 0.4823 \\ \vdots & \ddots & \vdots \\ 0.3725 & \cdots & 0 \end{bmatrix}$

Data-preprocessing enable each data (image) presenting in same range (0~1), which could optimize the gradient descent.

Q1: Model architecture—Dense neural layer



Q2: Feed forward



$\therefore$ parameters of each layer could be generalized:

$$Z^n = W^{n^T} A^{n-1}$$
$$A^n = f(Z^n)$$

.....

$$Z^1 = W^{1^T} X + b^1$$
$$A^1 = f(Z^1)$$

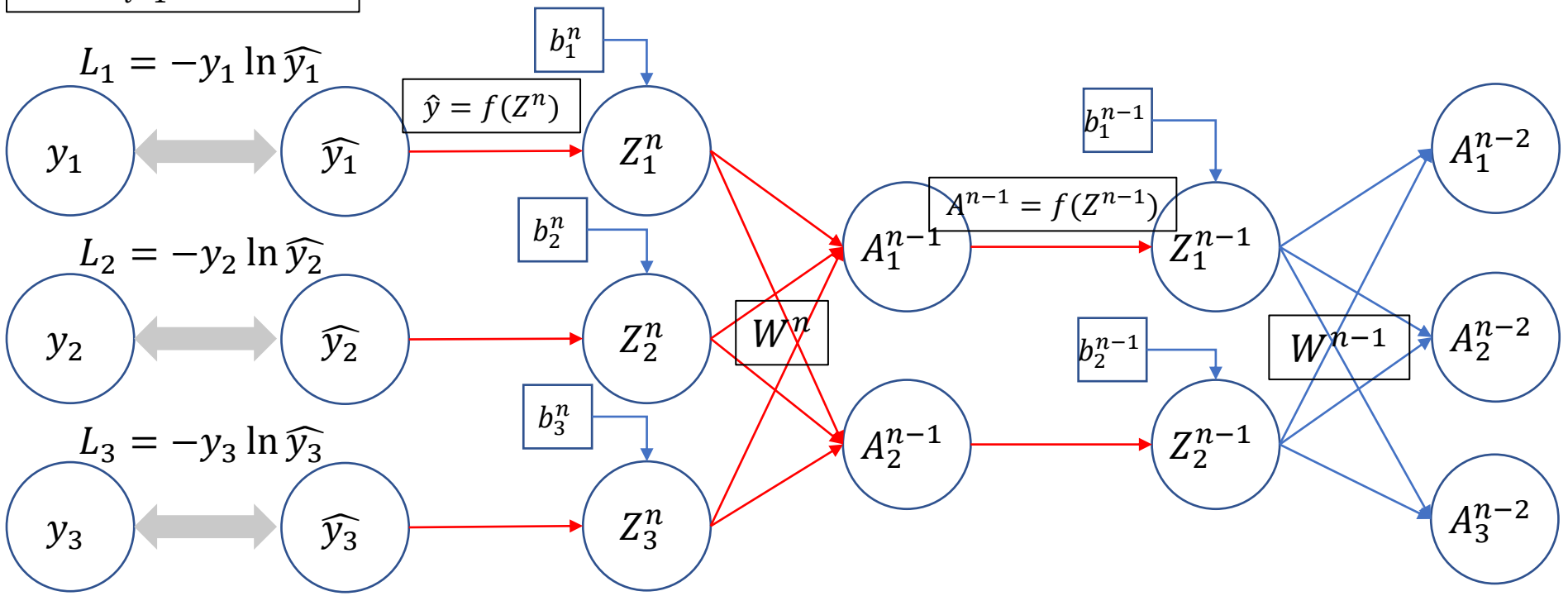
$$Z^2 = W^{2^T} A^1 + b^2$$
$$A^2 = f(Z^2)$$

Q2: Back propagation

Loss: Cross-entropy

$$-\sum_{i=1}^{n} y_i \ln \hat{y}_i$$

Target: solve $\frac{\partial L}{\partial W}, \frac{\partial L}{\partial b}$



1. $\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}}$

2. $\frac{\partial L}{\partial Z^n} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial Z^n} = \frac{\partial L}{\partial \hat{y}} f'(Z^n)$

3. $\frac{\partial L}{\partial W^n} = \frac{\partial L}{\partial Z^n} \frac{\partial Z^n}{\partial W^n} = \frac{\partial L}{\partial Z^n} A^{n-1}$

3. $\frac{\partial L}{\partial b^n} = \frac{\partial L}{\partial Z^n} \frac{\partial Z^n}{\partial b^n} = \frac{\partial L}{\partial Z^n}$

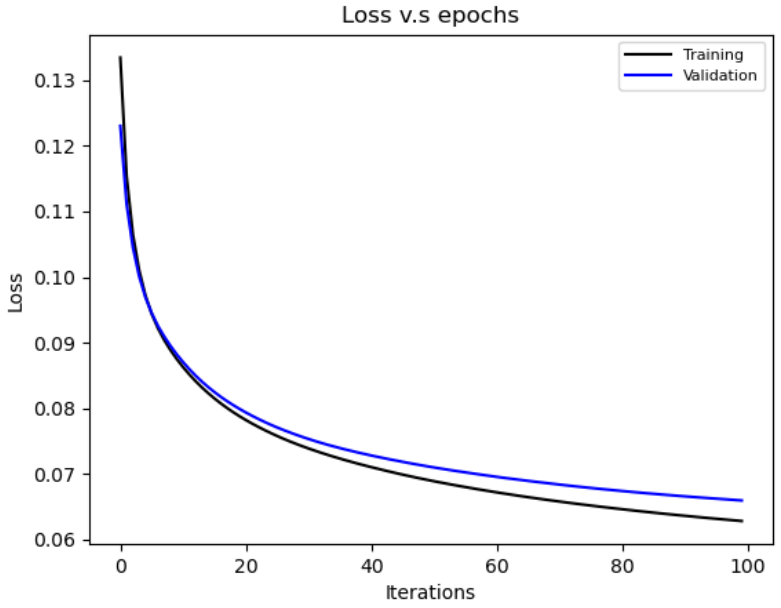
4. $\frac{\partial L}{\partial A^{n-1}} = \frac{\partial L}{\partial Z^n} \frac{\partial Z^n}{\partial A^{n-1}} = \frac{\partial L}{\partial Z^n} W^n$

5. $\frac{\partial L}{\partial Z^{n-1}} = \frac{\partial L}{\partial A^{n-1}} \frac{\partial A^{n-1}}{\partial Z^{n-1}} = \frac{\partial L}{\partial A^{n-1}} f'(Z^{n-1})$

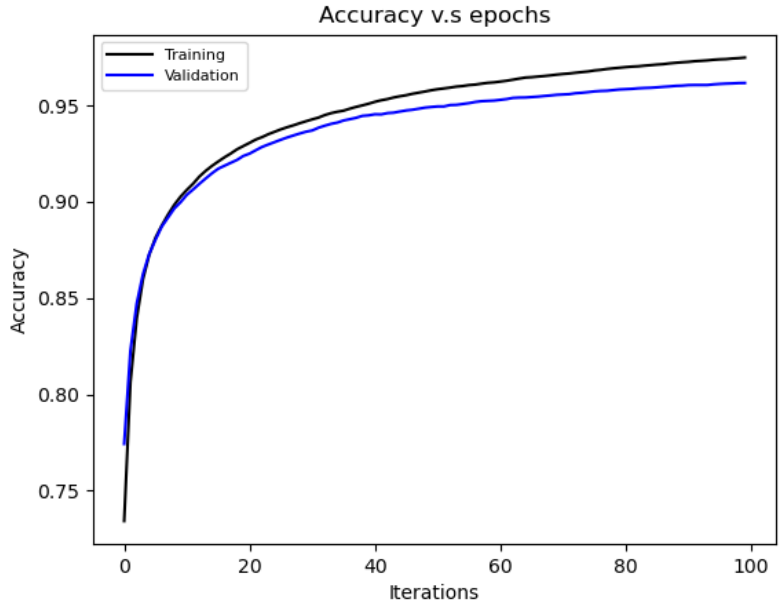
$\frac{\partial L}{\partial W^{n-1}} = \frac{\partial L}{\partial Z^{n-1}} \frac{\partial Z^{n-1}}{\partial W^{n-1}} = \frac{\partial L}{\partial Z^{n-1}} A^{n-2}$

$\frac{\partial L}{\partial b^{n-1}} = \frac{\partial L}{\partial Z^{n-1}} \frac{\partial Z^{n-1}}{\partial b^{n-1}} = \frac{\partial L}{\partial Z^{n-1}}$

Q3: Training Loss & Validation Loss



Loss after 100 epochs reduce to 0.065



Accuracy after 100 epochs reach 0.9617

Q4: If we use a very deep NN with a large number of neurons, will the accuracy increase?

No.

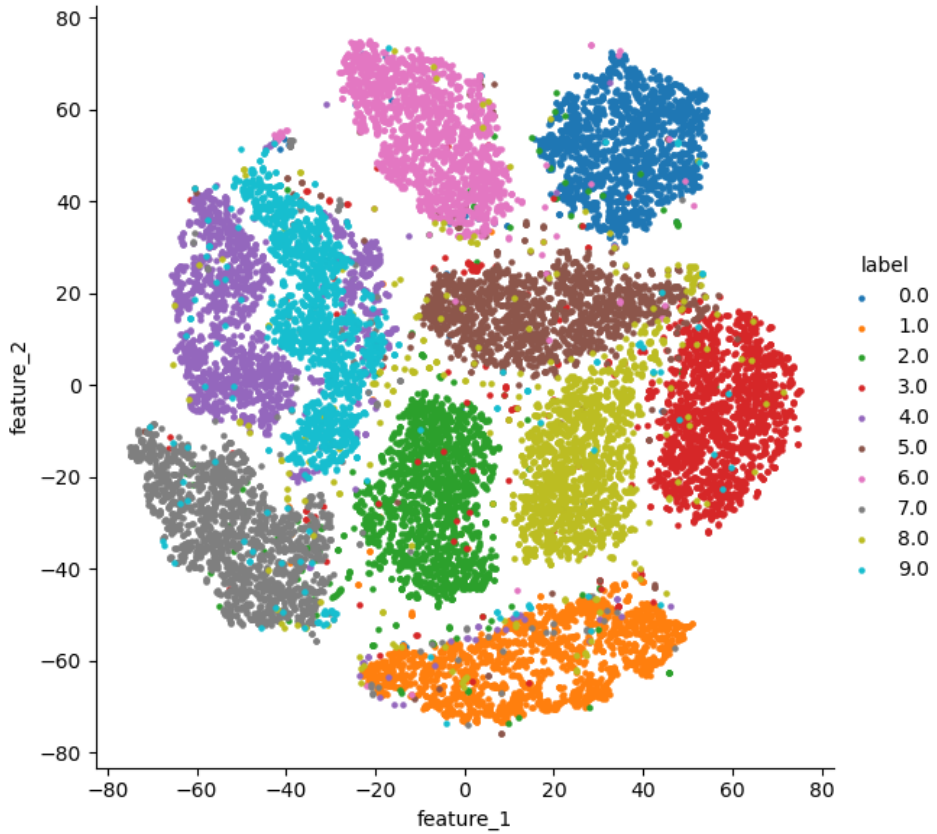
While deeper network and larger number of neurons could relatively increase the accuracy of prediction. But once the optimization is saturated, the deep structure may overfit the training data, and the testing accuracy would drop.

Q5: Why do we need to validate our model?

Validation data is used to estimate the model whether it is overfitted or not while model itself is still tuning the hyper-parameters. In other words, validation datasets give bias to the training model to avoid an unbiased evaluation in training data.

t-distributed stochastic neighbor embedding (T-SNE)

Labels of Validation data



Labels of testing data