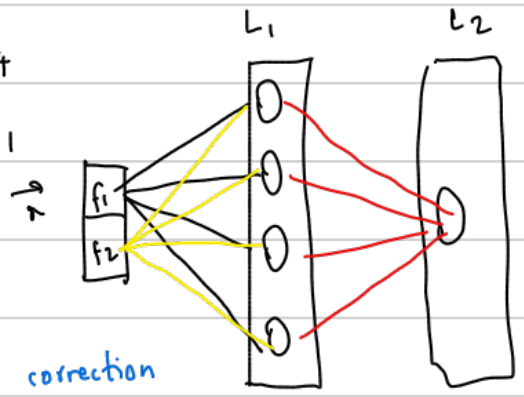
classification lossess → Binary classification loss - for classify two things.

layer1 ⇒ in:2 Out: 4

layer2 ⇒ in:4 out: 1



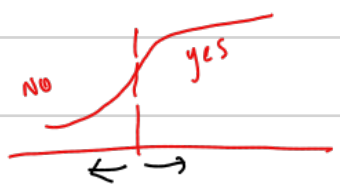
Data passing Forwarding

layer 01 → layer 02

} Also has Sigmoid

Sigmoid → The small correction

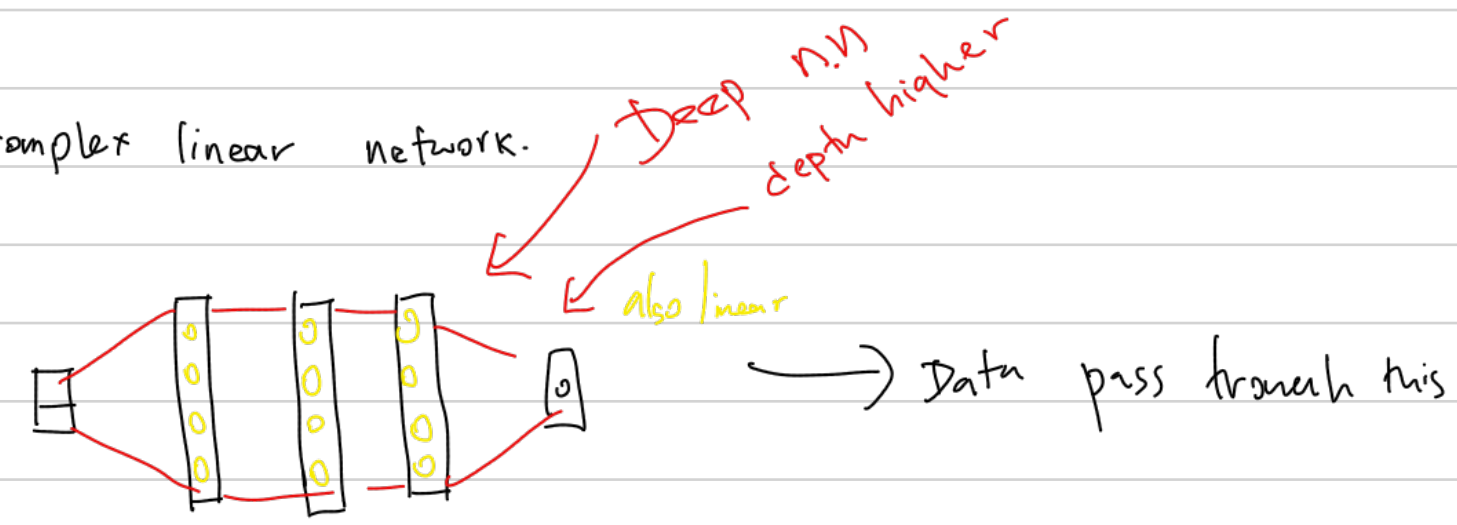
Activation Function.



Shallow layer

complex data points

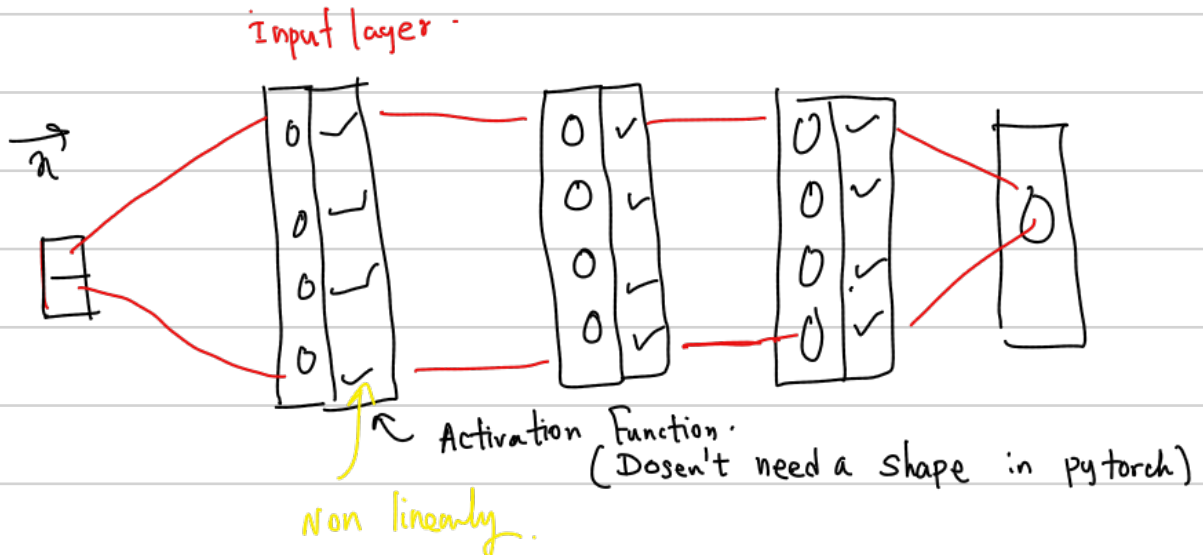
Try much more complex linear network.



→ Data pass through this

Network with None linearity ← "switching on and off nurones"

Deep learning with None linearty.



} Aproximate any function.

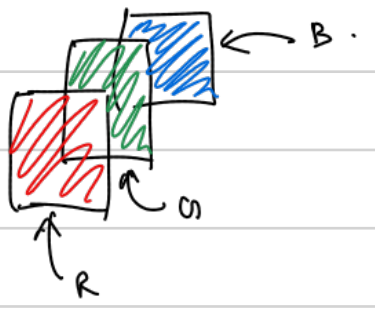
Universal Approximations

pytorch - Dataset - class.

- Init
- len
- getitem (idx)

Flattern is not good for images ? why ?

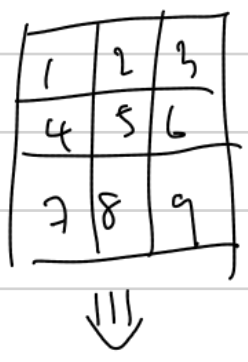
color image. : r g b. ← 3 channels



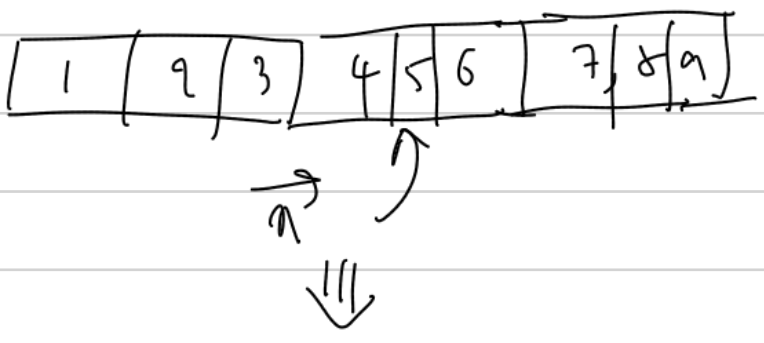
torch.Size([256, 1, 28, 28])

- 256 ← batch size
- 1 ← Chanells
- 28, 28 ← pixel 28*28.

feed to the Input layer



3 features.



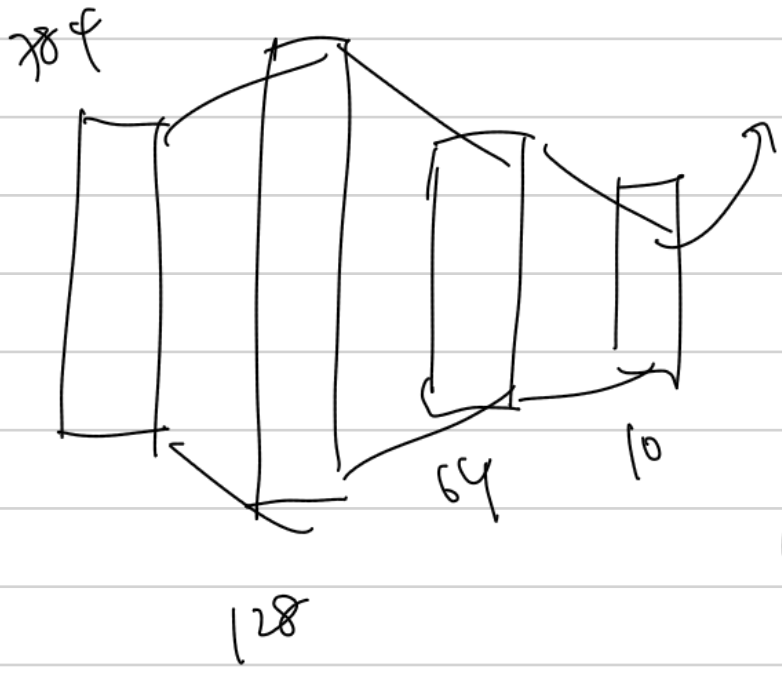
9 features.

Input layer. ← 784

Hidden layer. 1 ← 128 + ReLU

Hidden layer 2 ← 64 + Relu

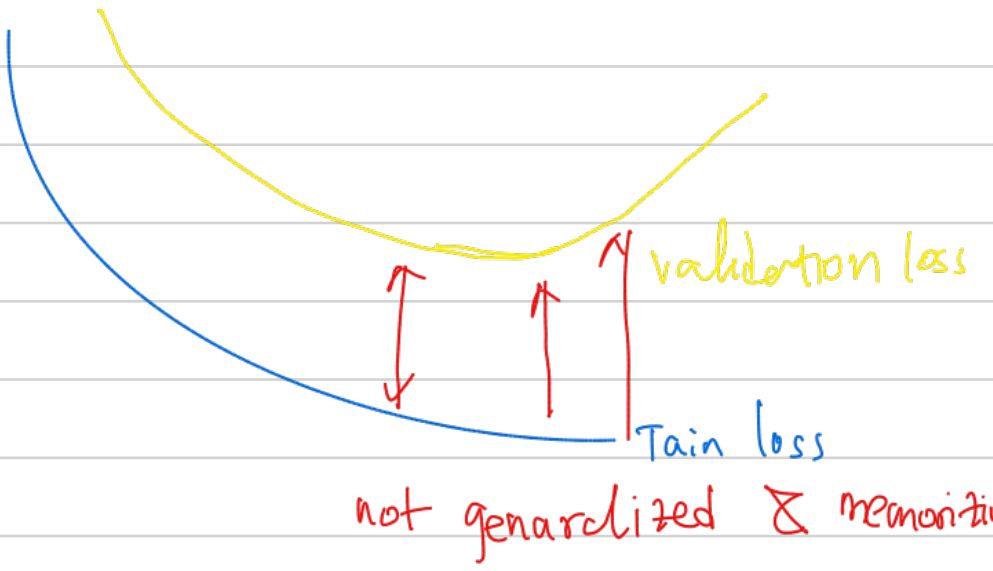
Output layer ← 10 (0-9)



- 0 — 0 ← make a probability
- 0 — 1
- 0 — 2
- 0 — 3
- 0 — 4
- 0 — 5
- 0 — 6
- 0 — 7
- 0 — 8
- 0 — 9

} multiclass

loss must be CrossEntropyLoss()



not generalized & memorizing