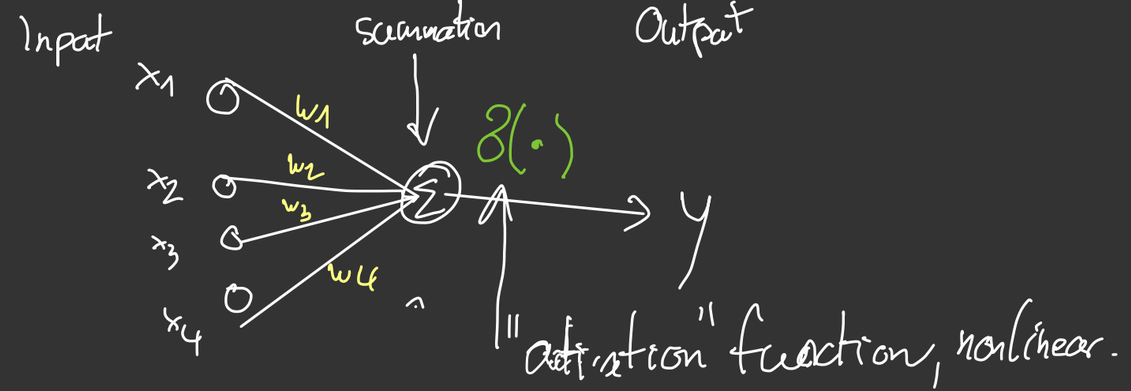
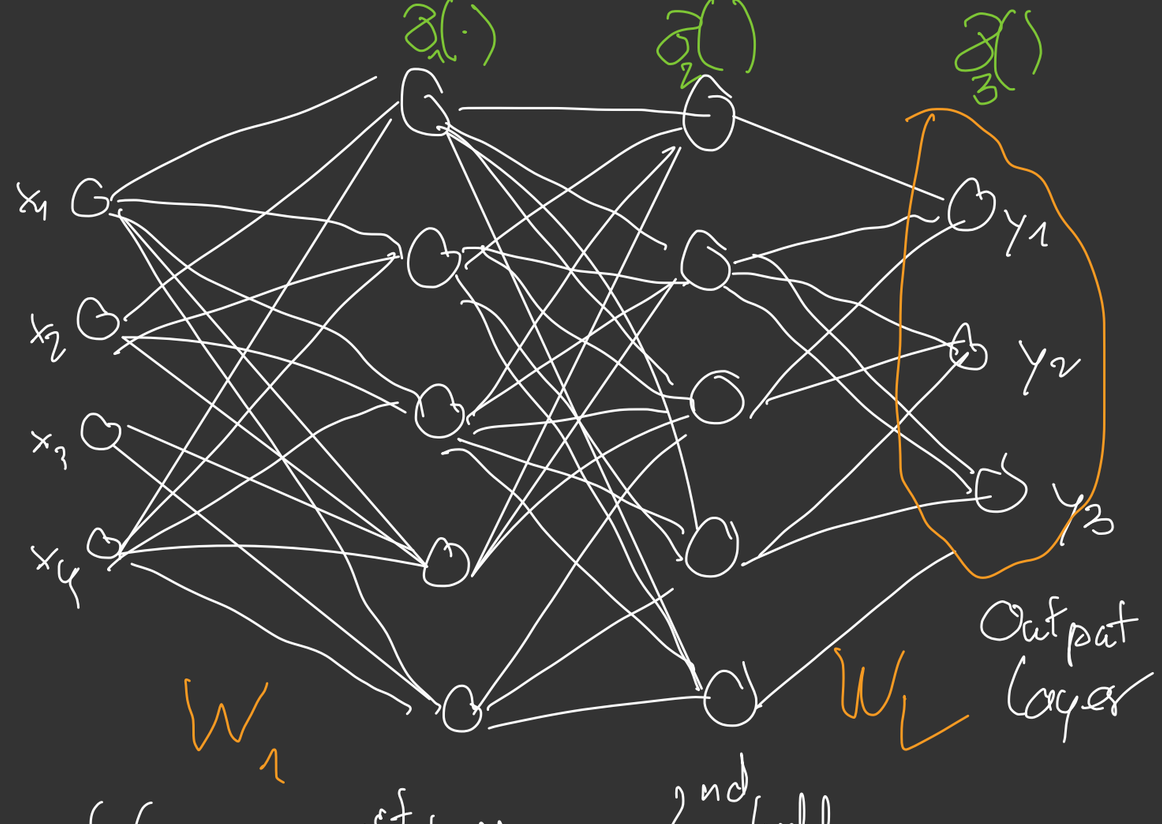
Artificial Neural Networks

(Selective) Little History: [McCullough, Pitts '43]: Mathematical model for neurons in brain:



- 1950's: Perceptron: 1st numerical scheme to use in classification
- 1980's: "Deeper" networks, successful training ... backpropagation
- Since ~2007: ▷ Efficient training, Regularization
  - ▷ Advances in computing technology: Use GPU's
  - ▷ Outperformance of traditional learning methods



A typical hypothesis space $\mathcal{F}$

width

$$\mathcal{F}_L = \{ h : X \to Y : h(x) = \sigma_L(W_L(\sigma_{L-1}(W_{L-1}(\dots \sigma_1(W_1(x)))))) \}$$

$\sigma_1, \dots, \sigma_L$ are non-linear, cts componentwise

$W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ (affine) Linear $\forall \ell = 1, \dots, L$

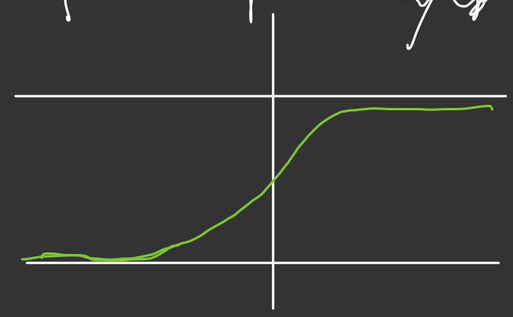
N_0 : dimension of input Layer

N_ℓ : — " — ℓ-th hidden Layer

N_L : — " — of output layer

Example: · Logistic Regression: 1-Layer NN ($L=1$) $\sigma(x) = \mathrm{Softmax}(x)$

· MLP with $L > 1$: Multilayer Perceptron

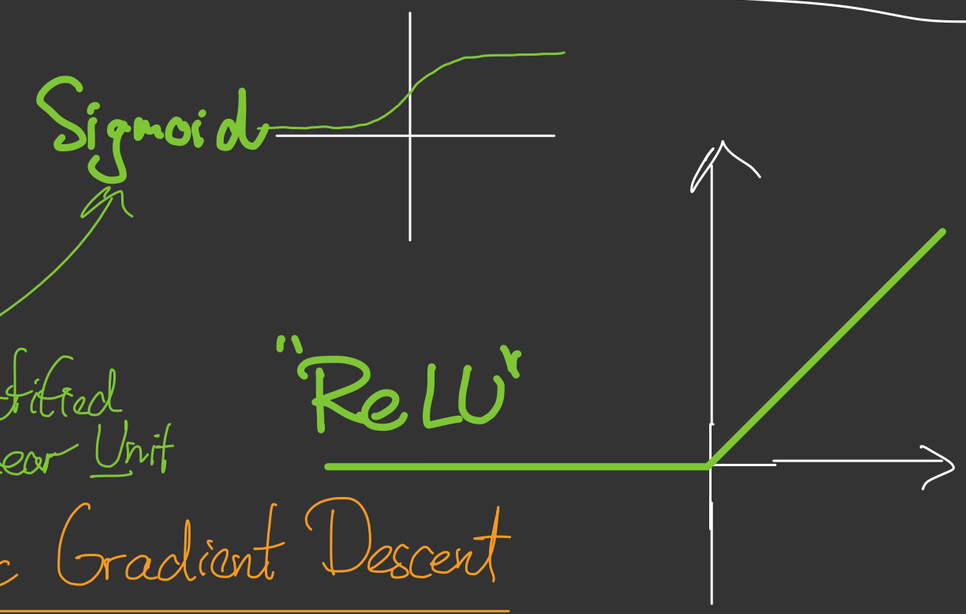


Thm: [Hornik, Cybenko]:

Every measurable function can be approximated by a neural network with at least $L=2$ layers (if wide enough).

Practical Question:

- How many layers L?
- Which loss function ℓ?
- Which activation function?
- How to "train"?



- Stochastic Gradient Descent
- Adam

Modifications / variants:

- Constrain weights to have certain properties: **Convolutional Neural Networks (CNN)**
  good for image problems (enforce spatial invariance).
- Fewer connections → less overfitting
- (Max/Average) Pooling:
  Reduces spatial sensitivity
- Dropout: Drops (at random) connections between layers in training phase
- Batch normalization: Normalizes input of a layer
  ↳ faster learning, better generalization

Epoch: One pass through entire training data,

Learning Rate: Stepsize of SGD