Last time:

- Backprop example
- Neural networks

Today: neural networks continued

Neural Network

\# of layers = "depth" of nn

- A type of function which is defined as a composition of "layers" (sub-functions)

  e.g. $f(X) = H_2(X, H_0(X), H_1(H_0(X)))$

- Each layer has a linear piece and a nonlinear "activation function."

- Several foundational types of layers we will cover
  - fully connected ①
  - convolutional ②
  - batch normalization ④
  - dropout ⑤
  - pooling ③
  - residual ⑥
  - attention ⑦

- the composition of many simple nonlinear layers builds a very expressive function

Fully-connected (Dense) Layer

→ linear piece:

$$\overset{[n\times d_k]}{Z_k(X)} = \overset{[n\times p]}{X}\ \overset{[p\times d_k]}{W_k} + \overset{[n\times d_k]}{B_k} \rightarrow \begin{bmatrix} — b_k — \\ \vdots \\ — b_k — \end{bmatrix}$$

only d_k free parameters

→ this layer is said to have d_k "neurons" (# columns of W_k)

→ intuition: each neuron represents a template or definition of a concept that is then searched for in the input X (via dot product)

→ number of parameters: $p \times d_k + d_k = (p+1)d_k$

Activation Functions

→ each layer is (usually) associated with an activation function (σ_k) which is applied after the linear piece ($Z_k(X)$).

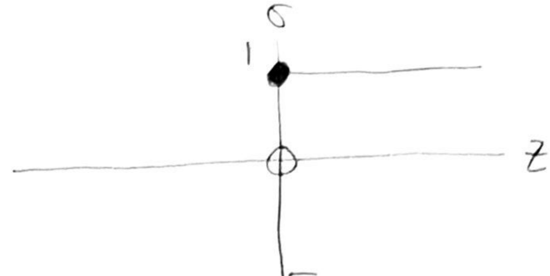
$$\overset{[n\times d_k]}{H_k(X)} = \sigma_k(\overset{[n\times d_k]}{Z_k(X)})$$

→ σ_k is (usually) applied to either each element of Z_k independently, or row-wise (e.g. softmax)

- **indicator / heaviside:**

$$\sigma(z) = \mathbb{1}\{z \geq 0\}$$

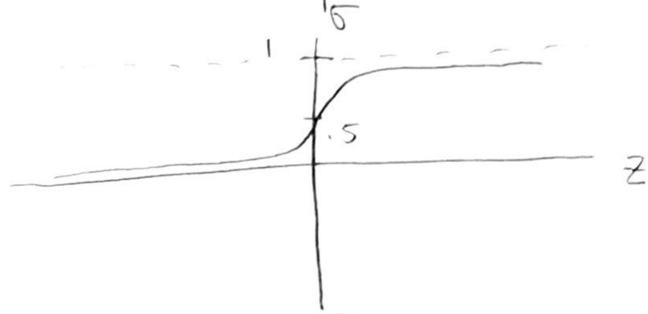
$$\sigma'(z) = 0$$



- **sigmoid:**

$$\sigma(z) = \frac{1}{1+\exp(-z)}$$

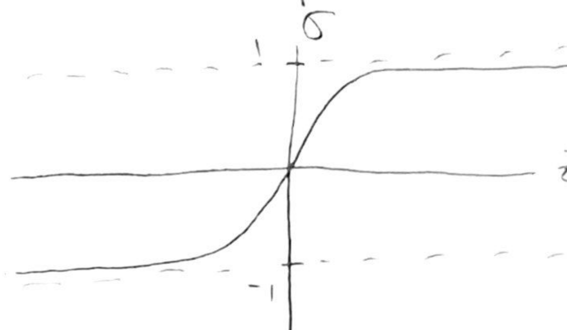
$$\sigma'(z) = (1-\sigma(z))\,\sigma(z)$$



- **tanh:**

$$\sigma(z) = \frac{\exp(2z)-1}{\exp(2z)+1}$$

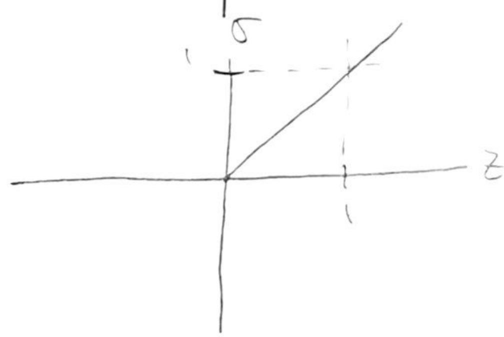
$$\sigma'(z) = 1 - [\sigma(z)]^2$$



- **rectified linear unit (relu):**

$$\sigma(z) = \max(0, z)$$

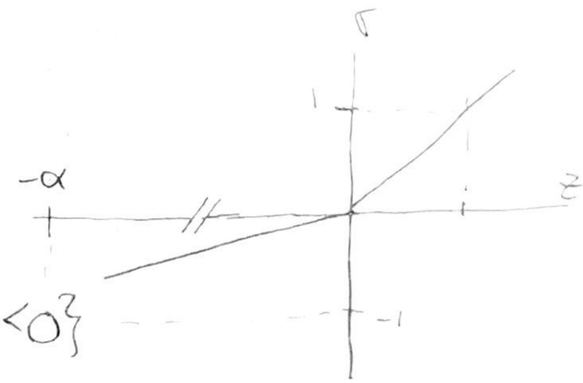
$$\sigma'(z) = \mathbb{1}\{z \geq 0\}$$



- **leaky relu:**

$$\sigma(z) = \max\left(z, \frac{z}{\alpha}\right)$$

$$\sigma'(z) = \mathbb{1}\{z \geq 0\} + \frac{1}{\alpha}\mathbb{1}\{z < 0\}$$



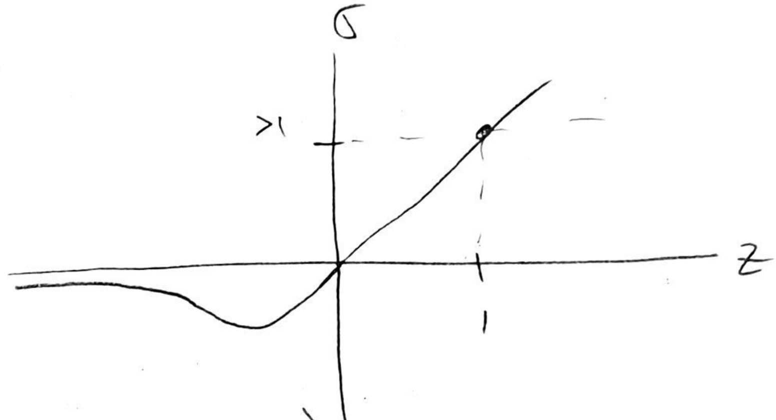
→ <u>gaussian error linear unit (gelu)</u>

$$\sigma(z) = z\,\Phi(z)$$

standard normal cdf

$$\sigma(z) \approx \frac{1}{2} z \left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(z + 0.044715 z^3\right)\right]\right)$$

$$\sigma(z) \approx z \cdot \text{sigmoid}(1.702 \cdot z)$$



$$\sigma'(z) \approx \frac{\exp\{\alpha z\}(\alpha z + \exp\{\alpha z\} + 1)}{(\exp\{\alpha z\} + 1)^2}, \quad \alpha = 1.702$$

→ <u>softmax (applied row-wise to Z)</u>

$$\sigma(z) : \mathbb{R}^d \to S^d, \quad S = \left\{ s \in \mathbb{R}_+^d : \sum_{j=1}^{d} s_j = 1, \; s_j \geq 0 \;\forall j \right\}$$

$$\sigma_j(z_i) = \frac{\exp\{z_{ji}\}}{\sum_{j=1}^{d} \exp\{z_{ji}\}}$$

eg. $z = [1, 2, 3, -1]$ $\quad \sigma(z) = [0.089, 0.242, 0.657, 0.012]$

→ can use to transform any vector into $\underline{\underline{a}}$ probability dstn.

Example: "feedforwad" (fully connected) n.n.

$X = [n \times p]$

$$\overset{[n \times d_3]}{f(X)} = \left[\max\left(0, \underbrace{\max(0, X\overset{[p\times d_1]}{W_1} + \overset{[n\times d_1]}{B_1})}_{H_1}\right)\overset{[d_1\times d_2]}{W_2} + \underset{[n\times d_2]}{B_2}\right]\overset{[d_2\times d_3]}{W_3} + \underset{[n \times d_3]}{B_3}$$

- batch size? n
- activation function? relu
- numbes of layers? 3 (2 hidden layers)
- output shape? $[n \times d_3]$
- number of parameters? $p = 9,\ d_1 = 16,\ d_2 = 8,\ d_3 = 1$

$$= [(p \times d_1) + d_1] + [(d_1 \times d_2) + d_2] + [(d_2 \times d_3) + d_3]$$

$$= [(9 \times 16) + 16] + (16 \times 8) + 8 + (8 \times 1) + 1$$

$$= [144 + 16] + [128 + 8] + [8 + 1]$$

$$= 160 + 136 + 9 = 305$$

- $f: \mathbb{R}^p \to \mathbb{R}^1$

Biological Analogy / Architecture Diagram

- Each layer has d_h neurons. Each neuron receives stimulus from the input X or neurons in preceding layers. Each neuron is either excited or not, and passes this information to neurons in succeeding layers.