Units in Neural Networks

CPT_S 434/534 Neural network design and application

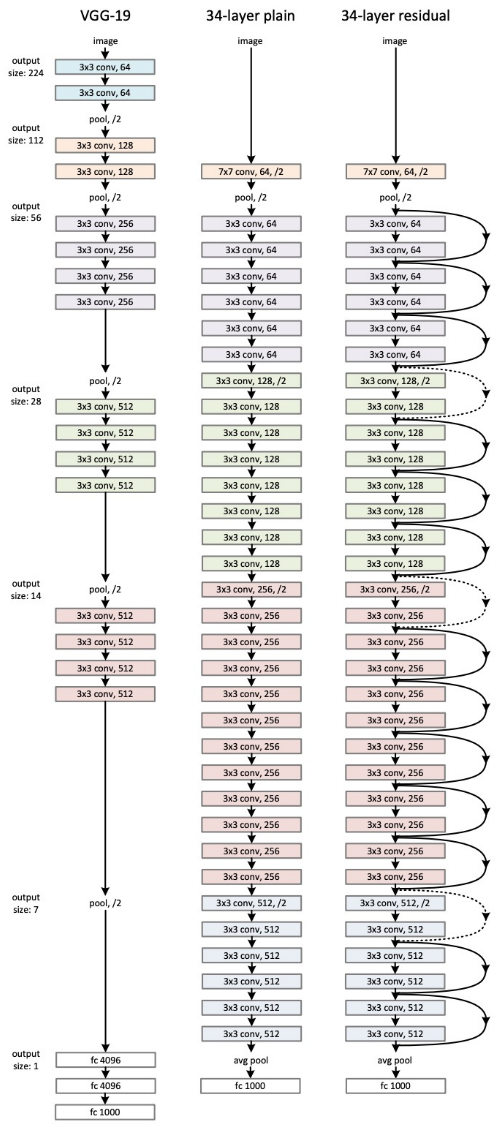
Previous

- NN basics
- An example of learning XOR function

Feedforward networks

- Deep:
  - Many compositional layers
- Nonlinearity
  - Some functions f_i can be nonlinear
- Nonconvexity
  - Composition of functions
  - Some functions f_i can be nonconvex
- Feedforward
  - Information feedforward from input to output layer

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) = \hat{y}_i$$



Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \quad \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}}$$

$$\boldsymbol{X}\boldsymbol{W} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$\boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

max(0, ·)

$$\boxed{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \xleftarrow{\boldsymbol{w}^\top} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Q: any explanation on the reason why we successfully learn this problem?

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{cc} x_1 & x_2 \\ \multicolumn{2}{c}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}} \end{array} \quad \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}} \longrightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

max(0, ·)

$$\boxed{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}} \xrightarrow{\boxed{\boldsymbol{w}}^\top} \boxed{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}}$$

Q: any explanation on the reason why we successfully learn this problem?

Weight in linear function

Learned feature (not 2d coordinates)

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boxed{\boldsymbol{w}}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + \cancel{b}}$

Learning XOR function

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \boxed{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}} \end{array} \quad \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}} \rightarrow \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \xrightarrow{+\boldsymbol{c}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boxed{\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}},$$

Original feature

$\max(0, \cdot)$

$$\boxed{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \xleftarrow{\boxed{\boldsymbol{w}}^\top} \boxed{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}}$$

Q: any explanation on the reason why we successfully learn this problem?

Weight in linear function

Learned feature (not 2d coordinates)

What if we use a nonlinear function as: $f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boxed{\boldsymbol{w}}^\top \boxed{\max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + \cancel{b}}$

In today's class

- Units for neural networks
  - Output units → cost function
    - $f_m\left(\ldots\left(f_2(f_1(w; x_i))\right)\right) \rightarrow y_i$

In today's class

- Units for neural networks
  - Output units → cost function
    - $f_m\left(\dots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$
  - Hidden units
    - $f_m\left(\dots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$

Cost function and output units

Binary	Gray code	One-hot
000	000	00000001
001	001	00000010
010	011	00000100
011	010	00001000
100	110	00010000
101	111	00100000
110	101	01000000
111	100	10000000

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Likelihood function

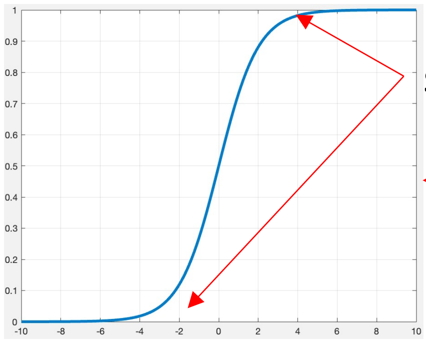
$$L(w) = P_w(X_1 = x_1, \dots, X_n = x_n) = f(w;x_1)\times\cdots\times f(w;x_n) = \prod_{i=1}^{n} \boxed{f(w;x_i)}$$

approximate

Probability mass function
(e.g., Bernoulli distribution)

- Maximum likelihood estimation

$$\max_w \log L(w) = \log\prod_{i=1}^{n} f(w;x_i) = \sum_{i=1}^{n}\log(\boxed{f(w;x_i)})$$



satisfies probability format

$$f_{sigmoid}(z) = \frac{1}{1+e^{-z}}$$

z: may be unbounded

Cost function and output units

$$f_m\left(\dots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow[\text{approximate}]{} p_{\text{model}}(\mathbf{x};\boldsymbol{\theta})$$

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$$

Probability for data

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \to y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$$

Probability for data

Q: Where?

One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ $\xleftarrow{\text{approximate}}$ $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$

Probability for data

One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\dots\left(f_2(f_1(w; x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it

One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

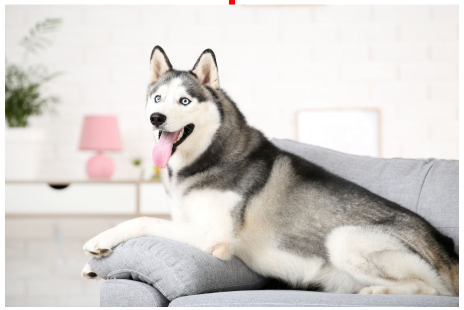
$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \to y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ ← approximate — $p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}})$ → $\boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it



One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction:

dog	cat	chair
0.98	0.01	0.01

$$\boxed{\boldsymbol{\theta}}$$

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}})$

Probability for data

Assume we know how to construct it

One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction:

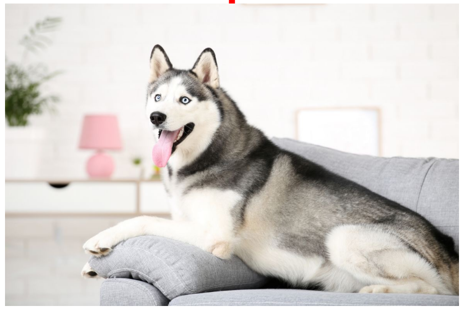
dog	cat	chair
0.98	0.01	0.01

$\boldsymbol{\theta}$

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}})$

Probability for data

Assume we know how to construct it



One-hot label:

dog	cat	chair
1	0	0

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it

Probability for "cat"

One-hot label:

	dog	cat	chair
	0	1	0

Cost function and output units

$$f_m\left(\dots\left(f_2(f_1(w;x_i))\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Probability for "cat"

One-hot label:

dog	cat	chair
0	1	0

Assume we know how to construct it

Cost function and output units

$$f_m\left(\ldots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

Probability prediction:

dog	cat	chair
0.01	0.98	0.01

$\boldsymbol{\theta}$

$p_{\text{data}}(\mathbf{x})$ approximate $p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}})$

Probability for data

Assume we know how to construct it

Probability for "cat"

One-hot label:

dog	cat	chair
0	1	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$$

Probability for data

Assume we know how to construct it

Probability prediction:

dog	cat	chair
0.01	0.98	0.01

Probability for “cat”

One-hot label:

dog	cat	chair
0	1	0

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \to y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x})$ ← approximate — $p_{\text{model}}(\mathbf{x}; \boxed{\boldsymbol{\theta}})$ → $\boxed{\boldsymbol{\theta}}$

Probability for data

Probability prediction for "dog/cat/chair" on all data

Assume we know how to construct it

One-hot labels for "dog/cat/chair" on all data

Cost function and output units

$$f_m\left(\dots\left(f_2\left(f_1(w;x_i)\right)\right)\right) \rightarrow y_i$$

- How to interact with groundtruth labels?
- Maximum likelihood estimation

$p_{\text{data}}(\mathbf{x}) \xleftarrow{\text{approximate}} p_{\text{model}}(\mathbf{x};\boxed{\boldsymbol{\theta}}) \rightarrow \boxed{\boldsymbol{\theta}}$

Probability for data

Assume we know how to construct it

Probability prediction for “dog/cat/chair” on all data

One-hot labels for “dog/cat/chair” on all data

Cost function and output units

- MLE and KL divergence

$$\min_{p_{model}} D_{\mathrm{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \log p_{\text{model}}(\boldsymbol{x})\right]$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} D_{\text{KL}}\left(\boxed{\hat{p}_{\text{data}}} \| p_{\text{model}}\right) = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \log p_{\text{model}}(\boldsymbol{x})\right]$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \log p_{\text{model}}(\boldsymbol{x})\right]$$

If we minimize the KL divergence between the two distributions:

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

$$= \boxed{\arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})}$$

Cost function and output units

- MLE and KL divergence

Empirical distribution (from training set): cannot scan all possible data $p_{\text{data}}(\mathbf{x})$

$$\min_{p_{model}} \boxed{D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right)} = \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \boxed{\log p_{\text{model}}(\boldsymbol{x})}\right]$$

If we minimize the KL divergence between the two distributions:

Equivalent to

$$\min_{p_{model}} -\mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \boxed{\left[\log p_{\text{model}}(\boldsymbol{x})\right]}$$

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

$$= \boxed{\arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})}$$

Minimizing KL divergence

=

MLE

What are hidden units?

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

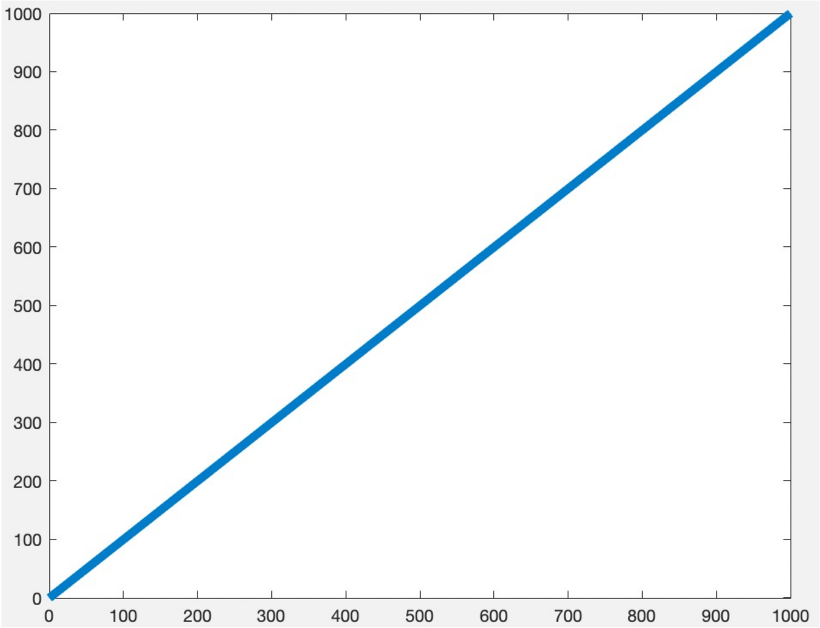
Inputs $\longrightarrow \boxed{f} \longrightarrow$ Outputs

What are hidden units?

$$f_m\left(\dots\left(f_2\left(f_1(w; x_i)\right)\right)\right) \rightarrow y_i$$

Inputs $\longrightarrow$ $\boxed{f}$ $\longrightarrow$ Outputs

Q: what if for all layers:
$f(x) = a * x$?

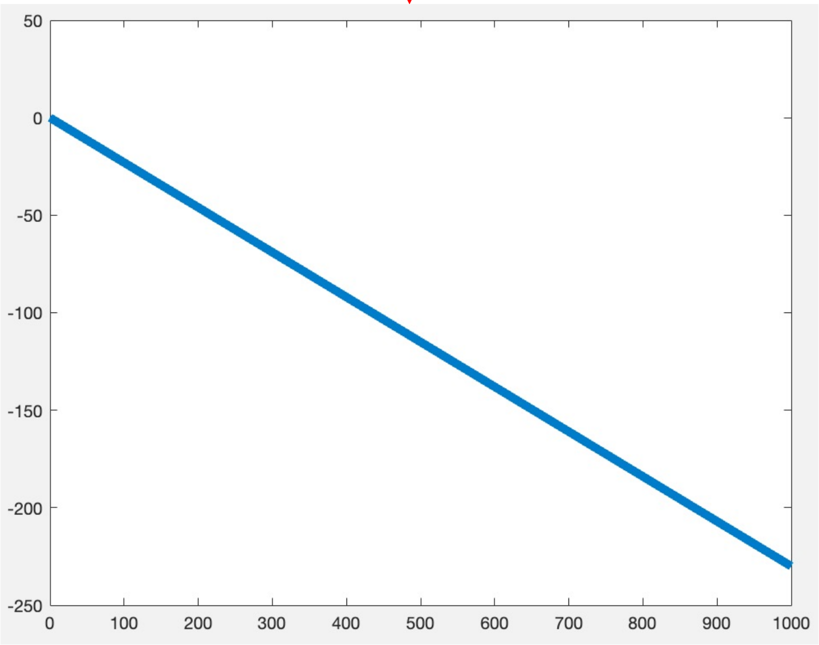


$f(x) = x$

What are hidden units?

$$f(x) = a * b * c * d * x$$

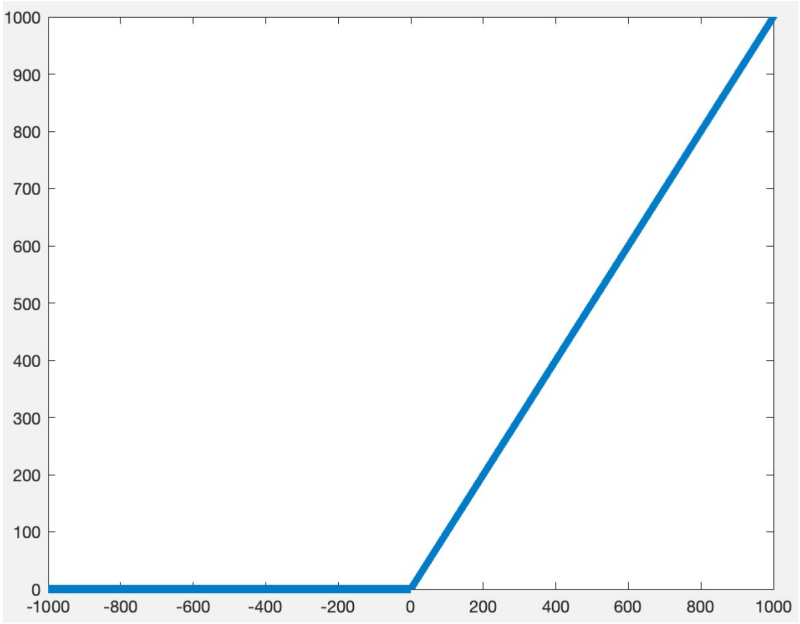
?



Combination of all linear layers is still linear
We are interested in nonlinear layers

ReLU (Rectified Linear Unit)

Activation function

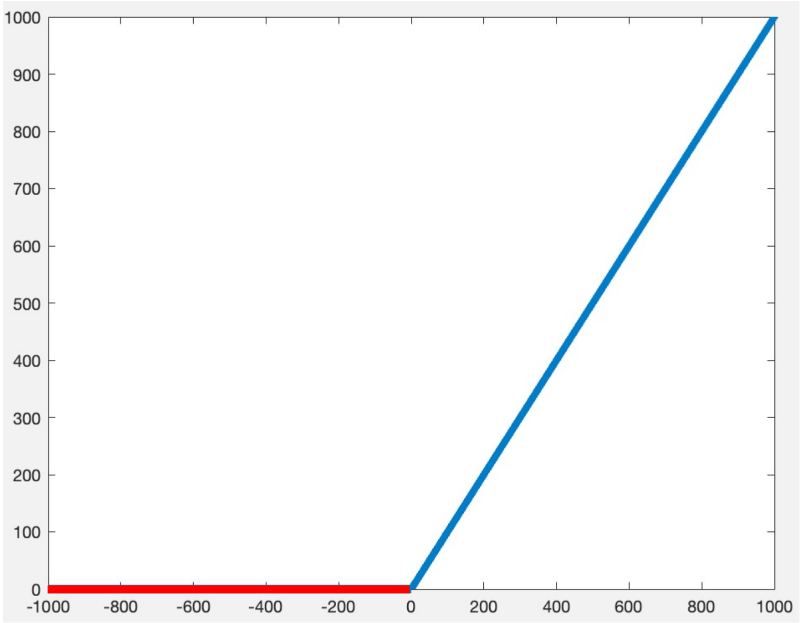


$$f(x) = \max(0, x)$$

ReLU (Rectified Linear Unit)

- Dying ReLU issue

Activation function



$$f(x) = \max(0, x)$$

Determining model parameters

- When to terminate GD (determining T)?
  - Main factors influencing convergence rate?
    - Step size (learning rate)

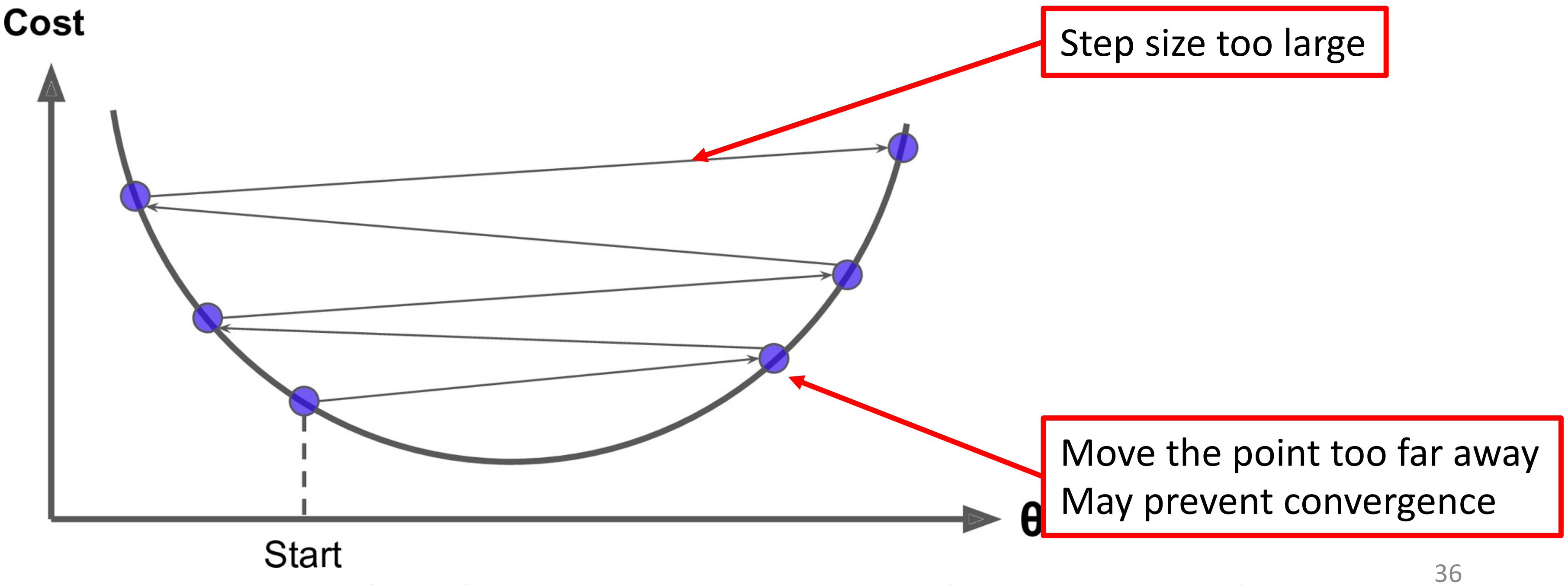
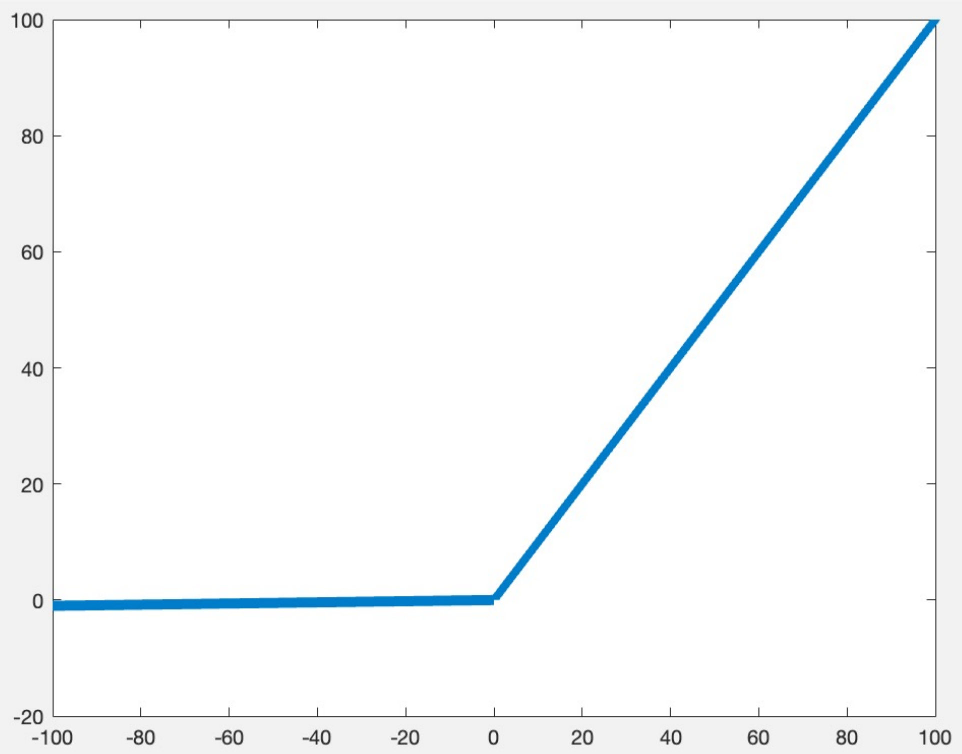


Image from https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/ch04.html

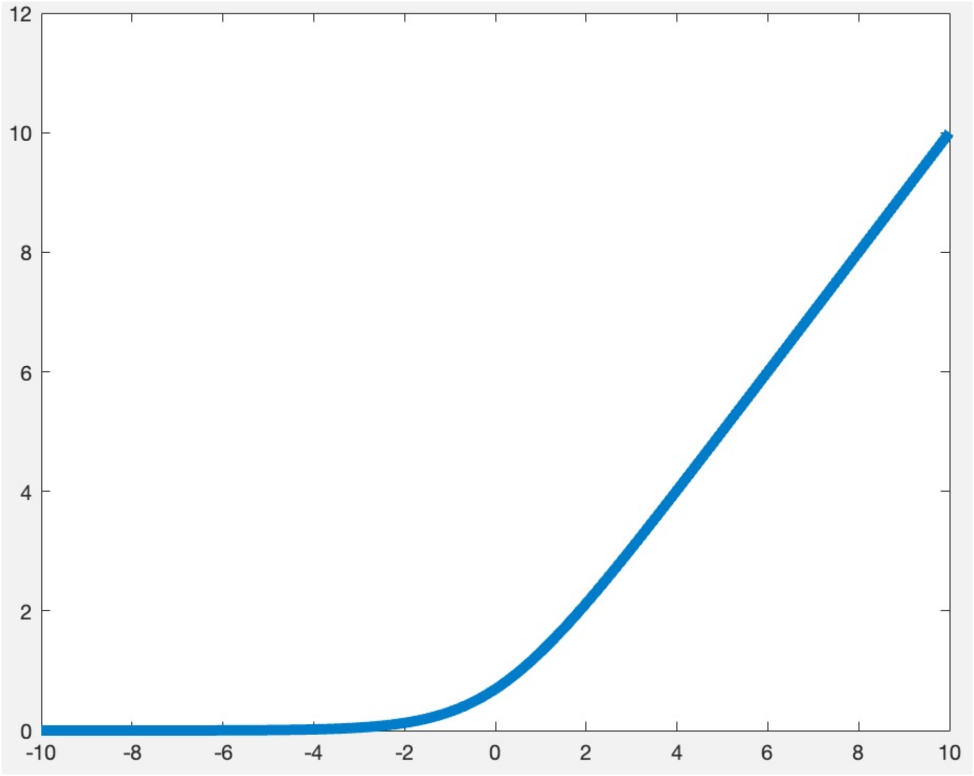
Leaky ReLU

$$f(x_j^i) = \max(0.01x_j^i, x_j^i)$$



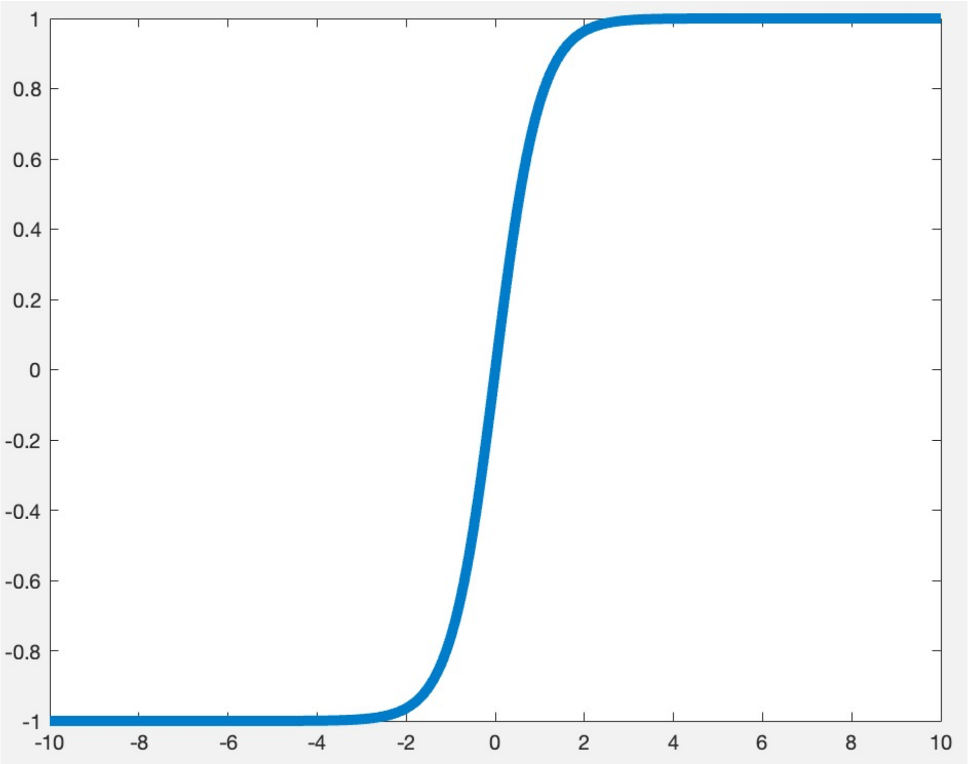
Smooth ReLU/softplus

$$a_j^i = f(x_j^i) = \log\left(1 + \exp(x_j^i)\right)$$



Tanh

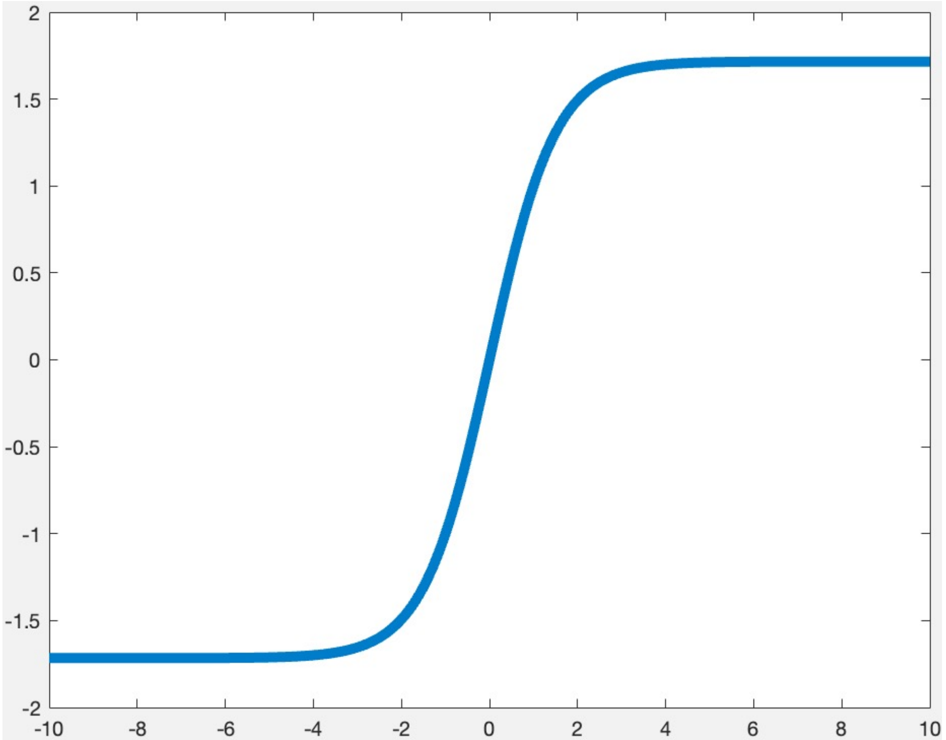
$$f(x_j^i) = \tanh(x_j^i)$$



Credit for https://adl1995.github.io/an-overview-of-activation-functions-used-in-neural-networks.html

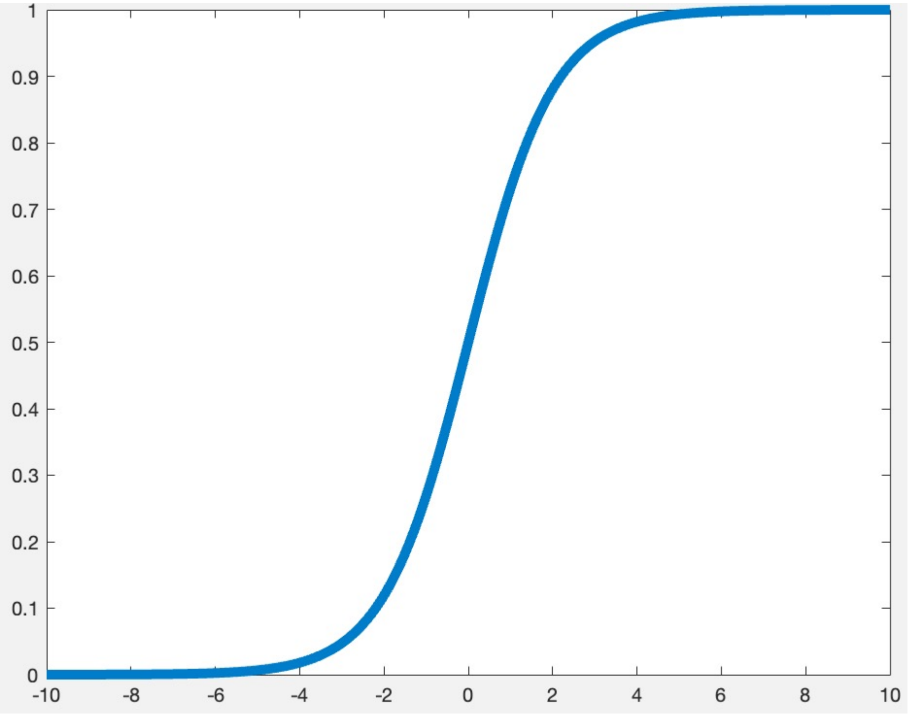
LeCun's Tanh [1]

$$f(x_j^i) = 1.7159 \tanh\left(\frac{2}{3}x_j^i\right)$$



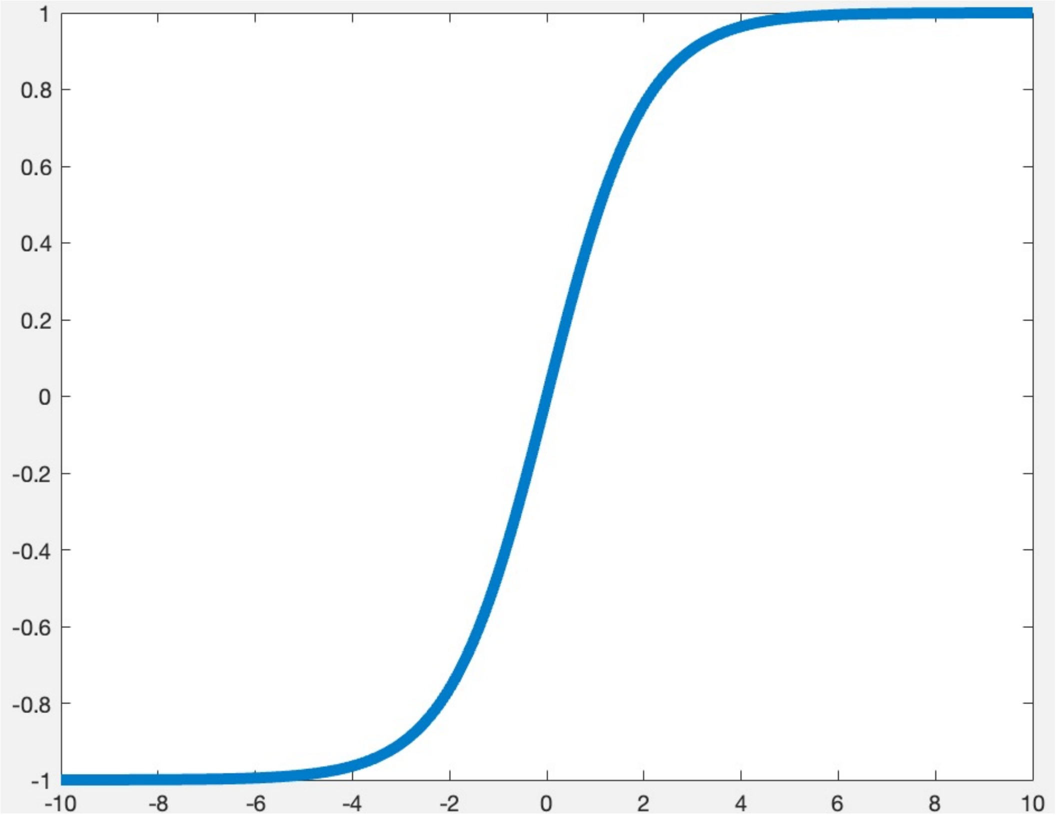
Sigmoid

$$f(x_j^i) = \frac{1}{1 + \exp(-x_j^i)}$$



Bipolar sigmoid

$$f(x_j^i) = \frac{1 - \exp(-x_j^i)}{1 + \exp(-x_j^i)}$$



References

- [1] LeCun, Yann A., Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. "Efficient backprop." In *Neural networks: Tricks of the trade*, pp. 9-48. Springer, Berlin, Heidelberg, 2012.

  Pdf online: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf