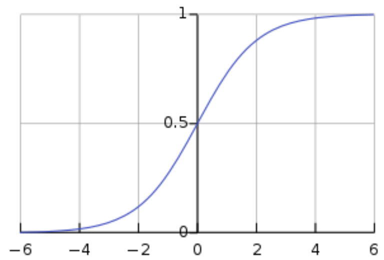
MT Summer Term 2021 Ex9: A Bluffer's Guide to NNs/Intro to NNs Ryan Harris

1. The sigmoid (σ, also called "the logistic") is a non-linear activation function:



Please give the mathematical definition of $\sigma(x)$ (using Euler's number *e*).

2. Given the definition requested in (1), what is the value of $\sigma(x)$ if:

I. $x \to \infty$
II. $x \to -\infty$
III. $x = 0$

Why is the sigmoid called a "squishification" function?

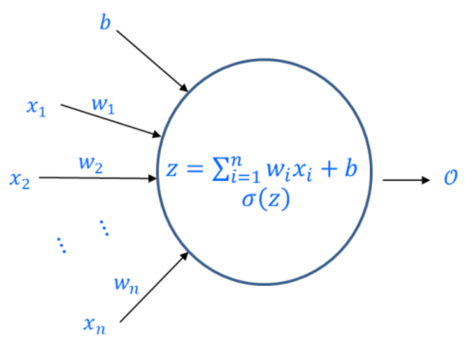
3. Please use

$c' = 0$	where c a constant
$(c\, x^n)' = c\, n\, x^{n-1}$	where c a constant
$(f \pm g)' = f' \pm g'$	sum rule
$(f \times g)' = f' \times g + f \times g'$	product rule
$\left(\frac{f}{g}\right)' = \frac{f' \times g - f \times g'}{g^2}$	quotient rule
$\left(f(g(x))\right)' = \left(f(z)\right)' \times z'$	chain rule, where $z = g(x)$
$(e^x)' = e^x$	exponential function

to show that $\sigma'(x) = (\sigma(x)(1 - \sigma(x))$

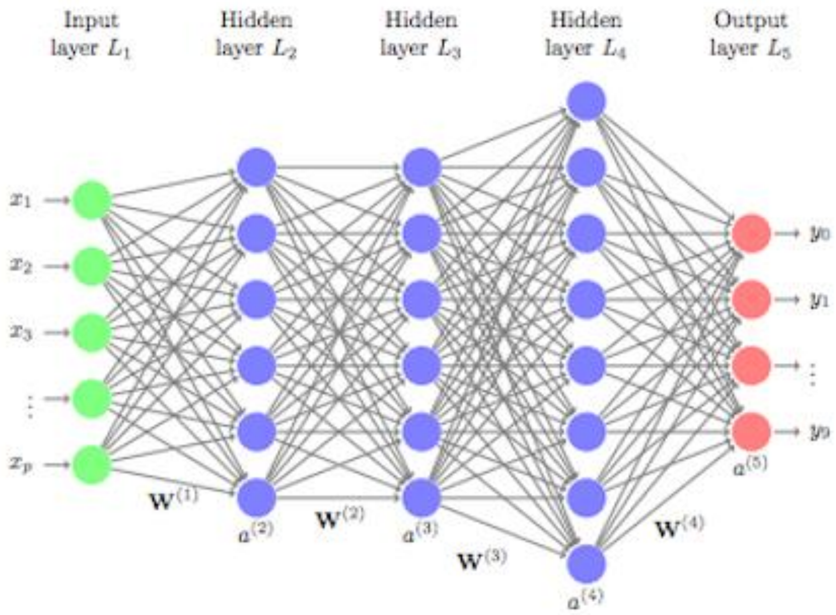
4. For what input x is the gradient $\sigma'(x)$ the steepest? Given that $\sigma'(x) = (\sigma(x)(1 - \sigma(x))$, what is the value of $\sigma'(x)$ at that point x?

5. Please show how you can absorb a bias term b of an AN into an input and a corresponding weight? Draw the corresponding picture and provide the corresponding equations.



$$\boxed{\begin{aligned} O &= \sigma(z) \\ &= \sigma\left(\textstyle\sum_{i=1}^{n} w_i x_i + b\right) \end{aligned}}$$

6. Given the following example of a simple feed forward neural network



and assuming that $x, a^{(2)}, a^{(3)}, a^{(4)}, a^{(5)}, y$ are all **row vectors** of dimensionality $x \in \mathbb{R}^{(1,p)}, a^{(2)} \in \mathbb{R}^{(1,6)}, a^{(3)} \in \mathbb{R}^{(1,6)}, a^{(4)} \in \mathbb{R}^{(1,8)}, a^{(5)} \in \mathbb{R}^{(1,9)}, y \in \mathbb{R}^{(1,9)}$ (where dimensionality is indicated $\mathbb{R}^{(rows,columns)}$).

Please provide the dimensionalities $\mathbf{W}^{(i)} \in \mathbb{R}^{(rows,columns)}$ for the weight matrices $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{W}^{(4)}$ such that the following hold:

$$x\,\mathbf{W}^{(1)} = a^{(2)}$$

$$a^{(2)}\mathbf{W}^{(2)} = a^{(3)}$$

$$a^{(3)}\mathbf{W}^{(3)} = a^{(4)}$$

$$a^{(4)}\mathbf{W}^{(4)} = a^{(5)}$$

Please provide the dimensionalities $\mathbf{W}^{(i)} \in \mathbb{R}^{(rows,columns)}$ for the weight matrices $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{W}^{(4)}$, such that the following hold:

$$\mathbf{W}^{(1)}x^{\mathrm{T}} = (a^{(2)})^{\mathbf{T}}$$

$$\mathbf{W}^{(2)}(a^{(2)})^{\mathrm{T}} = (a^{(3)})^{\mathrm{T}}$$

$$\mathbf{W}^{(3)}(a^{(3)})^{\mathrm{T}} = (a^{(4)})^{\mathrm{T}}$$

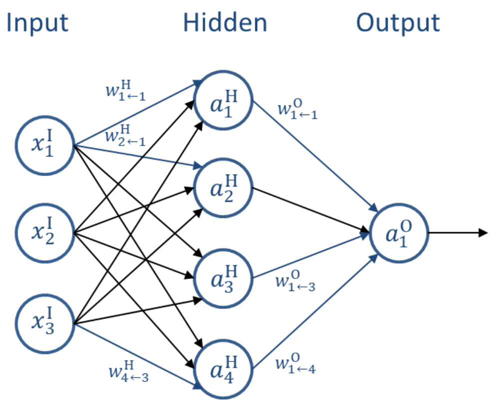
$$\mathbf{W}^{(4)}(a^{(4)})^{\mathrm{T}} = (a^{(5)})^{\mathrm{T}}$$

where Θ^{T} is the transpose of vector (or matrix) Θ.

7. Please use the notational conventions from the Richard Harris slides

- z_j^ℓ input to node j of layer ℓ
- $w_{j\leftarrow i}^\ell$ weight from layer $\ell - 1$ node i to layer ℓ node j
- $\sigma(z)$ sigmoid transfer function
- b_j^ℓ bias of node j in layer ℓ
- $\mathcal{O}_j^\ell$ output node j in layer ℓ
- t_j target value (ground truth) for node j in output layer

to vectorise:



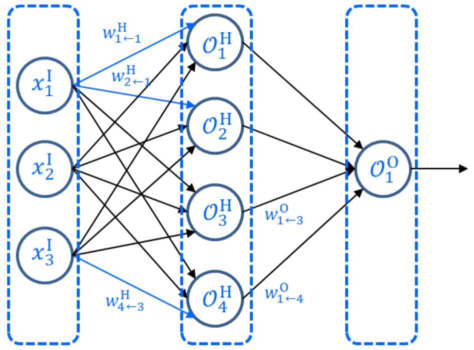
that is, assume that $\mathbf{x^I}$ is a column vector representing the input, $\mathbf{W^H}$and $\mathbf{W^O}$ weight matrices, $\mathbf{a^H}$ and $\mathbf{a^O}$ are column vectors representing the activations at the hidden and the output level of the FFNN such that $\sigma(\mathbf{W^H x^I}) = \mathbf{a^H}$ and $\sigma(\mathbf{W^O a^H}) = \mathbf{a^O}$. (Note that $\mathbf{a^O}$ is a "vector" with only one row and one column …). Please draw the vectors and matrices.

8. Please do the same as in (7), but where that $\mathbf{x^I}$ is a **row** vector representing the input, $\mathbf{W^H}$and $\mathbf{W^O}$ weight matrices, $\mathbf{a^H}$ and $\mathbf{a^O}$ are **row** vectors representing the activations at the hidden and the output level of the FFNN such that $\sigma(\mathbf{x^I W^H}) = \mathbf{a^H}$ and $\sigma(\mathbf{a^H W^O}) = \mathbf{a^O}$. Please draw the vectors and matrices.

9. Please do the same as in (8), but use transposes of the **column** vectors $\mathbf{x^I}$ in (7) to express the equations in (8). No need to draw the vectors and matrices.

10. What is the difference between a *loss*- vs. a *cost*-function in machine learning?

11. Please show that the partial derivative of the loss E with respect to a weight $w_{k\leftarrow j}^{\mathrm{O}}$ connecting to the output layer of the neural network



is: $\frac{\partial E}{\partial w^{\mathrm{O}}_{k\leftarrow j}} = (\mathcal{O}^{\mathrm{O}}_k - t_k)\mathcal{O}^{\mathrm{O}}_k(1-\mathcal{O}^{\mathrm{O}}_k)\mathcal{O}^{\mathrm{H}}_j$ Please label all the steps of the derivation with the partial differentiation rules used.

12. In your own words, please describe the Back-Propagation with stochastic Gradient Descent algorithm given below:

For each training instance:

1. Forward pass: compute prediction from input (also called inference, when model is fully trained)
2. For each output node compute: $\delta^{\mathrm{O}}_k = \mathcal{O}^{\mathrm{O}}_k(1-\mathcal{O}^{\mathrm{O}}_k)(\mathcal{O}^{\mathrm{O}}_k - t_k)$
3. For each hidden node compute: $\delta^{\ell-1}_j = \mathcal{O}^{\ell-1}_j\left(1-\mathcal{O}^{\ell-1}_j\right)\sum_{k\in\mathrm{O}}\delta^{\ell}_k\, w^{\ell}_{k\leftarrow j}$
4. For each weight:
   - Compute weight update: $\triangle\, w^{\ell}_{k\leftarrow j} = -\alpha\,\delta^{\ell}_k\,\mathcal{O}^{\ell-1}_j = -\alpha\,\frac{\partial E}{\partial w^{\ell}_{k\leftarrow j}}$
   - Update weight: $w^{\ell}_{k\leftarrow j} \coloneqq w^{\ell}_{k\leftarrow j} + \triangle\, w^{\ell}_{k\leftarrow j}$

   End

End

13. In your own words, what is the difference between

- Back-propagation with *stochastic* Gradient Descent
- Back-propagation with *mini-batch* Gradient Descent
- Back-propagation with *batch* Gradient Descent