Group	Name	8	9	10	11	12	13	$\sum$
	Svetlana Seliunina							
	Aleksei Zhuravlev							

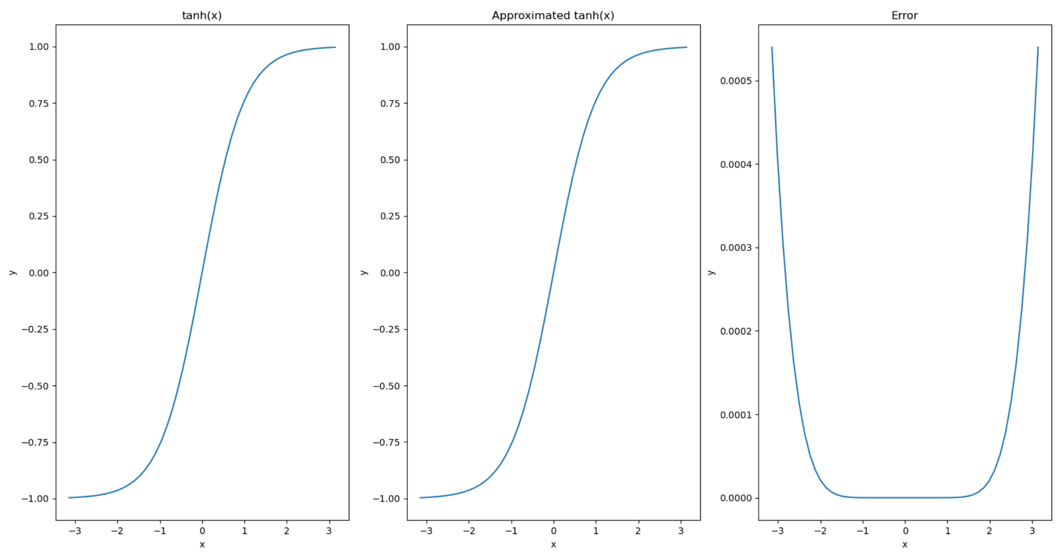
Technical Neural Networks Assignment Sheet 2

October 2022

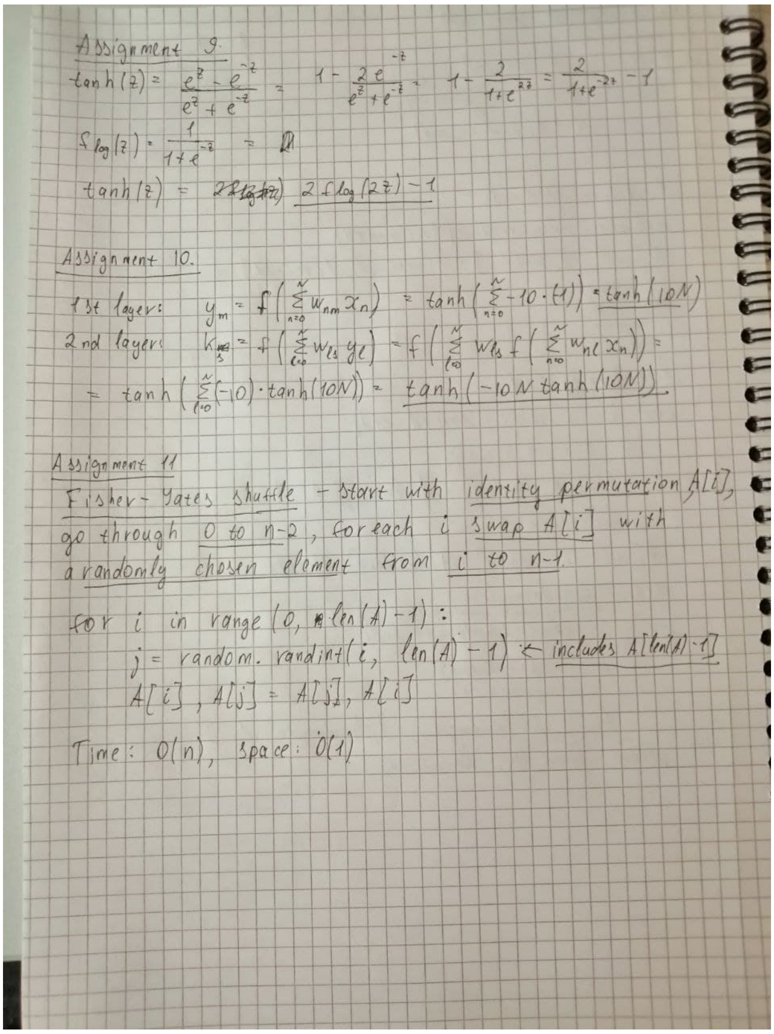
Assignment 8

We can write $\tanh(x)$ using Pade approximation:

$$\tanh x = \frac{x}{1 + \frac{x^2}{3 + \frac{x^2}{5 + \frac{x^2}{7 + \frac{x^2}{9}}}}}$$



Assignment 9



<u>Assignment 9.</u>

$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = 1 - \frac{2e^{-z}}{e^{z} + e^{-z}} = 1 - \frac{2}{1+e^{2z}} = \frac{2}{1+e^{-2z}} - 1$

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

$\tanh(z) = \underline{2 f_{log}(2z) - 1}$

<u>Assignment 10.</u>

1st layer: $y_m = f\left(\sum_{n=0}^{N} w_{nm} x_n\right) = \tanh\left(\sum_{n=0}^{N} -10 \cdot (-1)\right) = \underline{\tanh(10N)}$

2nd layer: $k_s = f\left(\sum_{l=0}^{N} w_{ls} y_l\right) = f\left(\sum_{l=0}^{N} w_{ls} f\left(\sum_{n=0}^{N} w_{nl} x_n\right)\right) =$

$= \tanh\left(\sum_{l=0}^{N} (-10) \cdot \tanh(10N)\right) = \underline{\tanh(-10N \tanh(10N))}$

<u>Assignment 11</u>

Fisher-Yates shuffle - start with identity permutation A[i], go through 0 to n-2, for each i swap A[i] with a randomly chosen element from i to n-1.

```
for i in range(0, len(A)-1):
    j = random.randint(i, len(A) - 1)  <- includes A[len(A)-1]
    A[i], A[j] = A[j], A[i]
```

Time: O(n), space: O(1)

Assignment 10

The 7 steps of the Backpropagation of Error Algorithm - Initialization

- Pick a pattern p, $\left({}^{p}X, pY\right)$

- Forward through the net, produce ${}^{p}Y$

- Compare with teacher $\left(p_y - {}^{p}y\right)$

- Backward through the net

- Calculate δ_m at output layer, and all Δw_{hm}

- Calculate δ_h at hidden layer, and all Δw_{kh}

- (repeat for all layer until input layer is reached)

- Update all weights, apply Δw_{ij}, $w_{ij} += \Delta w_{ij}$

- Ready? Stop if a reasonable criterion is met

Assignment 11

Assignment 13

$$ {}^{p}E^{**} = \underbrace{\frac{1}{2}\sum_{m=1}^{n} \left({}^{p}\hat{y}_m - {}^{p}y_m\right)^2}_{E_1^*} + \underbrace{\beta \frac{1}{2} \sum_{ij} (\omega_{ij})^2}_{E_2^*} $$

output layer:

$$ \frac{\partial E^{**}}{\partial \omega_{im}} = \frac{\partial E_1^*}{\partial \omega_{im}} + \frac{\partial E_2^*}{\partial \omega_{im}} $$

$$ \frac{\partial E_1^*}{\partial \omega_{im}} = -(\hat{y}_m - y_m)\, f_m'(net_m)\, \tilde{y}_i $$

$$ \frac{\partial E_2^*}{\partial \omega_{im}} = \frac{\partial \left(\beta \frac{1}{2} \sum_{ij} (\omega_{ij})^2\right)}{\partial \omega_{im}} = \beta \omega_{im} $$

$$ \Delta \omega_{im} = \eta \underbrace{(\hat{y}_m - y_m) f_m'(net_m)}_{\delta_m^{**}} \tilde{y}_i - \eta \beta \omega_{im} $$

$$ \Delta \omega_{im} = \eta\, \delta_m^{**}\, \tilde{y}_i - \eta \beta \omega_{im} $$

for hidden layer:

$$ \frac{\partial E^{**}}{\partial \omega_{im}} = \frac{\partial E_1^*}{\partial \omega_{im}} + \frac{\partial E_2^*}{\partial \omega_{im}} $$

$$ \frac{\partial E_1^*}{\partial \omega_{im}} = \left(\sum_{k=1}^{K} \delta_k^{**} w_{hk}\right) f_h'(net_m)\, \tilde{y}_i = \delta_m^{**} \cdot \tilde{y}_i $$

$$ \frac{\partial E_2^*}{\partial \omega_{im}} = \beta \omega_{im} $$

$$ \Delta \omega_{im} = \eta\, \delta_m^{**}\, \tilde{y}_i - \eta \beta \omega_{ih} $$

$$ \delta_m^{*} = \left(\sum_{k=1}^{K} \delta_k^{**} \omega_{mk}\right) f_m'(net_m) $$

Assignment 12

Programming assignment:

https://colab.research.google.com/drive/1JgDFgYZdTINlNhsn5OlzOsf7wEkGrRwH?usp=sharing