Group	Name	1	2	3	4	5	6	7	$\sum$
	Svetlana Seliunina								
	Aleksei Zhuravlev								

Technical Neural Networks Assignment Sheet-1

October 2022

Assignment 1

1. A lot of simple, interconnected processing units (neurons).
2. Modifiable, plastic connections (synapses, synaptic weights).
3. Mapping from an input vector to a scalar output value.
4. Nonlinear transfer function.

Assignment 2

1. Beginning: Early days when basic neural models and learning rules inspired by biological systems where discovered.
2. Quite Years: Lack of funding triggered by the proof that perzeptron type network cannot learn Boolean function XOR.
3. Renaissance: New multi-layer models and new learning rules caused rise of interest.
4. Consolidation: Neural network become widely used and accepted.
5. Second start-up: Discoveries of modern paradigms and applications.

Assignment 3

$$\Delta w_{ji} = \alpha \left(t_j - y_j\right) g' \left(h_j\right) x_i$$

where

α is a small constant called learning rate

$g(x)$ is the neuron's activation function

g' is the derivative of g

t_j is the target output

h_j is the weighted sum of the neuron's inputs, $h_j = \sum x_i w_{ji}$

y_j is the actual output,$y_j = g\left(h_j\right)$

x_i is the i th input.

Assignment 4

Because layers in MLP are fully connected, number of weights for a 4-layer MLP can be calculated using the following formula:

$$|w_4| = |w_{nh1}| + |w_{h1h2}| + |w_{h2m}| = (n+1)h_1 + (h_1+1)h_2 + (h_2+1)m,$$

where +1 indicates BIAS. For 3-layer MLP we get:

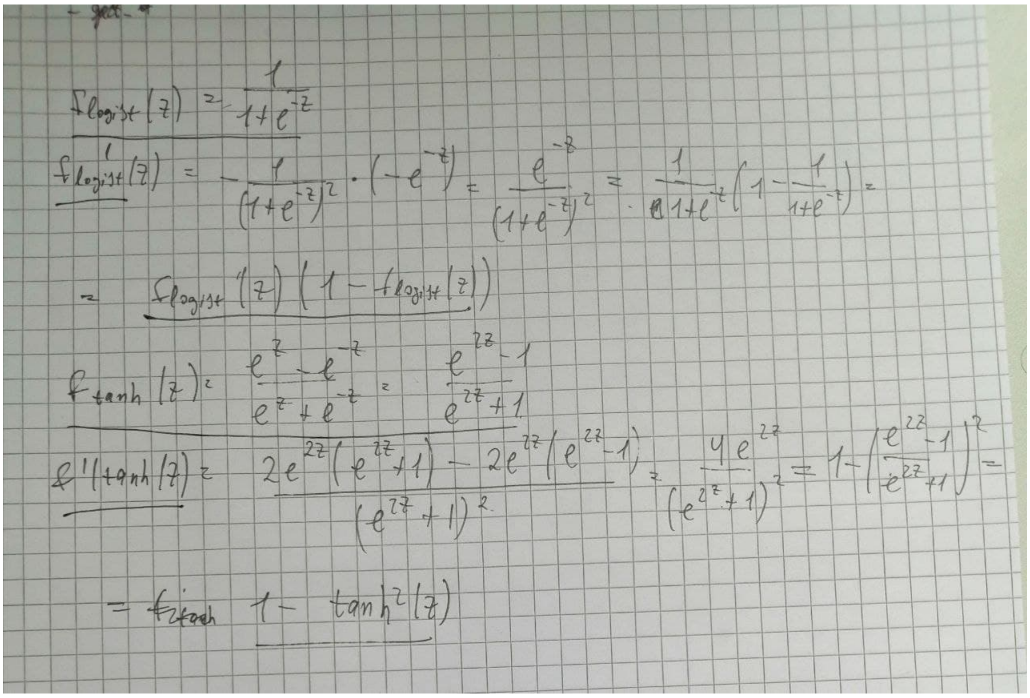
$$|w_3| = |w_{nh}| + |w_{hm}| = (n+1)h + (h+1)m = h(n+m+1) + m.$$

If $|w_4| \approx |w_3$ and $n = 5, h_1 = 21, h_2 = 30, m = 4$:

$$6 \cdot 21 + 22 \cdot 30 + 31 \cdot 4 = 10h + 4$$

$$h = 90.6 \approx 91$$

Assignment 5



Assignment 6

Proof. According to slide 12 of the second lecture, perzeptron can be described as a function $x_2 = -w_1/w_2 \cdot x_1 - w_0/w_2$ separating the 2D-plane in two parts. This function is linear so we need to prove that XOR Boolean function is not linearly separable. We will rewrite this line as

$L : a \cdot x_1 + b \cdot x_2 + c = 0.$
If two points lie on the same side of L then $(a \cdot x_{11} + b \cdot x_{21} + c)(a \cdot x_{12} + b \cdot x_{22} + c) > 0$, and if on opposites sides then $(a \cdot x_{11} + b \cdot x_{21} + c)(a \cdot x_{12} + b \cdot x_{22} + c) < 0$.

1. Points $(0, 0)$ and $(1, 1)$ lie on the same side: $c(a + b + c) > 0$.
2. Points $(1, 0)$ and $(1, 1)$ lie on the same side: $(a + c)(a + b + c) < 0$.
3. Points $(0, 1)$ and $(1, 1)$ lie on the same side: $(b + c)(a + b + c) < 0$.

Now we add (2) and (3) together:

$$(a + b + c)(a + b + 2c) = (a + b + c)^2 + c(a + b + c) < 0.$$

Because $(a + b + c)^2 \geq 0$, $c(a + b + c) < 0$ which contradicts (1). L cannot separate this function and therefore, a simple perzeptron without a hidden layer is not capable to implement the Boolean function XOR. □

Assignment 7

Error minimization by gradient descent: $\Delta w_{ij} = -\mu \frac{\partial E^*(w_{ij})}{\partial w_{ij}}$ where $E^* = \frac{1}{2} \sum_{m=0}^{M} (\hat{y_m} - y_m)^4$. We will start with the rule for output neuron and skip computations that are similar to those given in the lecture.

$$\frac{\partial E^*(w_{hm})}{\partial w_{hm}} = \frac{\partial E^*(w_{hm})}{\partial net_m} \cdot \frac{\partial net_m}{\partial w_{hm}} = \frac{\partial E^*(w_{hm})}{\partial y_m} \cdot \frac{\partial y_m}{\partial net_m} \cdot \frac{\partial net_m}{\partial w_{hm}}$$

According to the lecture:

$$\frac{\partial y_m}{\partial net_m} = f'(net_m); \frac{\partial net_m}{\partial w_{hm}} = \tilde{out}_h$$

Partial derivative of the error function:

$$\frac{\partial E^*(w_{hm})}{\partial y_m} = \frac{1}{2} \cdot 4 \cdot (\hat{y_m} - y_m)^3 \cdot (-1) = -2(\hat{y_m} - y_m)^3$$

Collecting all together:

$$\Delta w_{hm} = 2\mu(\hat{y_m} - y_m)^3 f'(net_m)\tilde{out}_h$$

$$\delta_m = -\frac{\partial E^*(w_{hm})}{\partial net_m} = 2(\hat{y_m} - y_m)^3 f'(net_m)$$

$$\Delta w_{hm} = \mu \delta_m \tilde{out}_h$$

Rule for hidden neuron:

$$\Delta w_{gh} = \mu \delta_h \tilde{out}_g$$

$$\delta_h = -\frac{\partial E^*(w_{gh})}{\partial net_h} = -\frac{\partial E^*(w_{gh})}{\partial out_h} \cdot \frac{\partial out_h}{\partial net_h} = -\frac{\partial E^*(w_{gh})}{\partial out_h} f'(net_h) =$$

$$= \sum_{k=1}^{K} \underline{\delta_k \mathrm{w}_{hk}} f'(net_h)$$

Derivation is the same as for Backpropagation of Error because it does not directly depend on partial derivative of E.

Assignment 8

Programming assignment:

`https://colab.research.google.com/drive/1P6k0W6upviiZiNSYfYHwvJtnJvXHpsyz?usp=sharing`