과제 1

[제목: 뉴럴네트워크]

과 목 명	머신러닝 응용
교 수 명	안 재 목
학 번	20237107
작 성 자	하 태 영
제 출 일	2025.11.24

한림대학교

문제 1

1. Layer1 출력(h_1): $h_1 = W_1x + b_1$
2. Layer2 출력(h_2): $h_2 = W_2h_1 + b_2$
3. Layer3 출력(h_3): $h_3 = W_3h_2 + b_3$

h_1 대입 → $h_2 = W_2(W_1x + b_1) + b_2 = W_2W_1x + W_2b_1 + b_2$

h_2 대입 → $h_3 = W_3(W_2W_1x + W_2b_1 + b_2) + b_3 = \underbrace{W_3W_2W_1}_{W_{eff}}x + \underbrace{W_3W_2b_1 + W_3b_2 + b_3}_{b_{eff}}$

$y = W_{eff}x + b_{eff}$ 선형 함수로 표현됩니다.

∴ 깊이와 상관없이 모델은 직선적인 결정 경계만 생성할 수 있으며, 이는 선형 분류기의 정의와 같습니다.

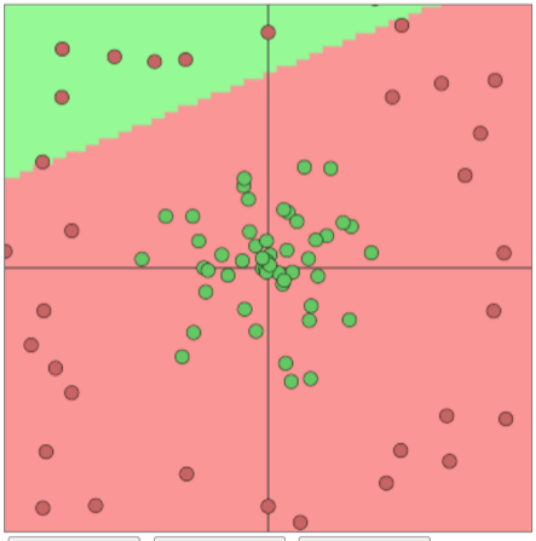
문제 2

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:2});
layer_defs.push({type:'fc', num_neurons:6});
layer_defs.push({type:'fc', num_neurons:2});
layer_defs.push({type:'softmax', num_classes:2});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1, batch_size:10, l2_decay:0.001});
```

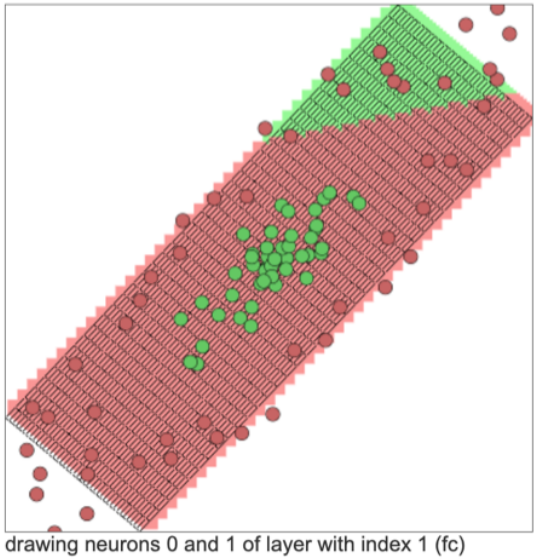
change network

Feel free to change this, the text area above gets eval()'d when you hit the button and the network gets reloaded. Every 10th of a second, all points are fed to the network multiple times through the trainer class to train the network. The resulting predictions of the network are then "painted" under the data points to show you the generalization.

On the right we visualize the transformed representation of all grid points in the original space and the data, for a given layer and only for 2 neurons at a time. The number in the bracket shows the total number of neurons at that level of representation. If the number is more than 2, you will only see the two visualized but you can cycle through all of them with the cycle button.



simple data | circle data | spiral data | random data



drawing neurons 0 and 1 of layer with index 1 (fc)

fc(6) | fc(2) | fc(2)

cycle through visualized neurons at selected layer (if more than 2)

Controls:
CLICK: Add red data point
SHIFT+CLICK: Add green data point
CTRL+CLICK: Remove closest data point

1. 결정 경계가 곡선이어야 좋은데 직선입니다.
2. 오분류된 점들이 매우 많습니다.

∴ 직선 경계와 다수의 오분류 포인트는 시각적 증거를 통해 비선형 분류가 필요함을 알 수 있습니다.

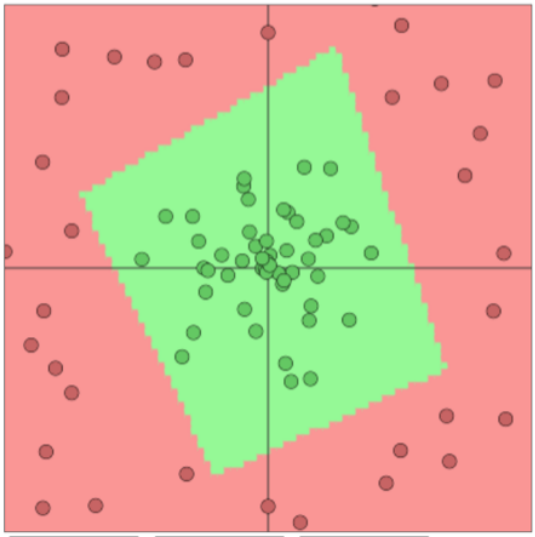
문제 3

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:2});
layer_defs.push({type:'fc', num_neurons:3, activation: 'relu'});
layer_defs.push({type:'softmax', num_classes:2});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1, batch_size:10, l2_decay:0.001});
```

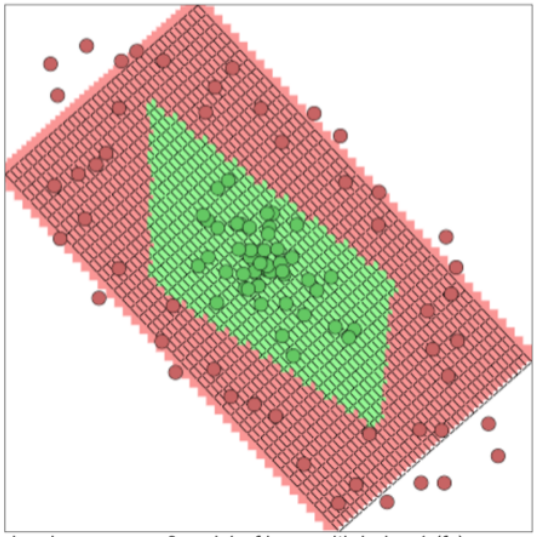
change network

Feel free to change this, the text area above gets eval()'d when you hit the button and the network gets reloaded. Every 10th of a second, all points are fed to the network multiple times through the trainer class to train the network. The resulting predictions of the network are then "painted" under the data points to show you the generalization.

On the right we visualize the transformed representation of all grid points in the original space and the data, for a given layer and only for 2 neurons at a time. The number in the bracket shows the total number of neurons at that level of representation. If the number is more than 2, you will only see the two visualized but you can cycle through all of them with the cycle button.



simple data | circle data | spiral data | random data



drawing neurons 0 and 1 of layer with index 1 (fc)

fc(3) | relu(3) | fc(2)

cycle through visualized neurons at selected layer (if more than 2)

Controls:
CLICK: Add red data point
SHIFT+CLICK: Add green data point
CTRL+CLICK: Remove closest data point