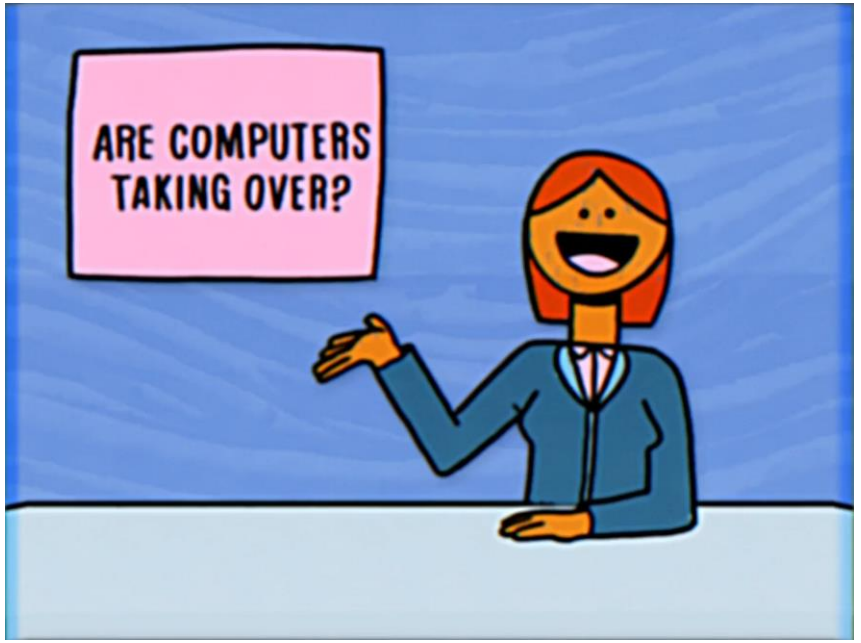
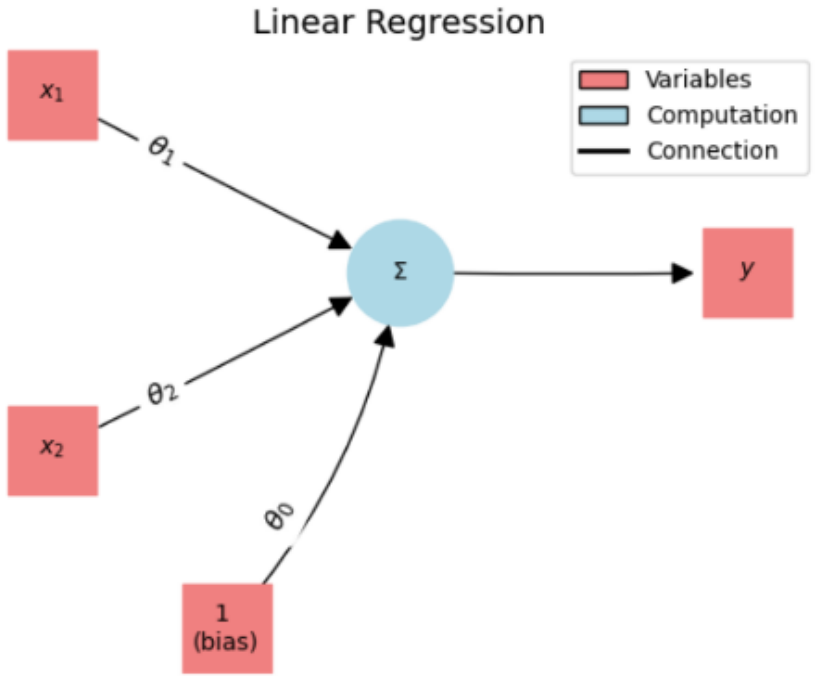
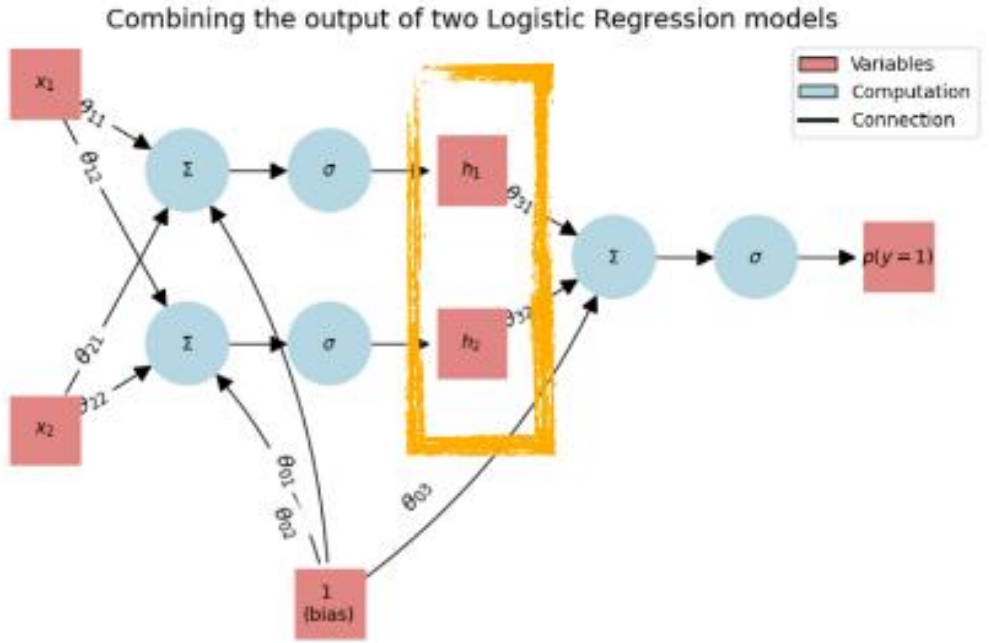
Neural Networks part 1

Myrthe Boone



From 1 linear regression → complexer models! → layers & learned features!





$$y = \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2$$

Q: advantage of learned features?

Describe logistic regression in ‘layers’

```
class LogisticRegressionModel(nn.Module):

    # START EXERCISE 2.1 #
    #'''
    def __init__(self, input_dim=2):

        super(LogisticRegressionModel,
self).__init__()
        self.linear = nn.Linear(input_dim, 1)
        self.output = nn.Sigmoid()

    def forward(self, x):

        logits = self.linear(x)
        p = self.output(logits)
        return p
```

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

Why complexer models?

- More features & more combinations of features → images!
- Sigmoid layer usually at the end → why?
- Does this sigmoid layer have any trainable parameters itself?
- Classification → why binary cross-entropy instead of MSE?
- Answer: maximizing **probabilities** (not minimizing distances)

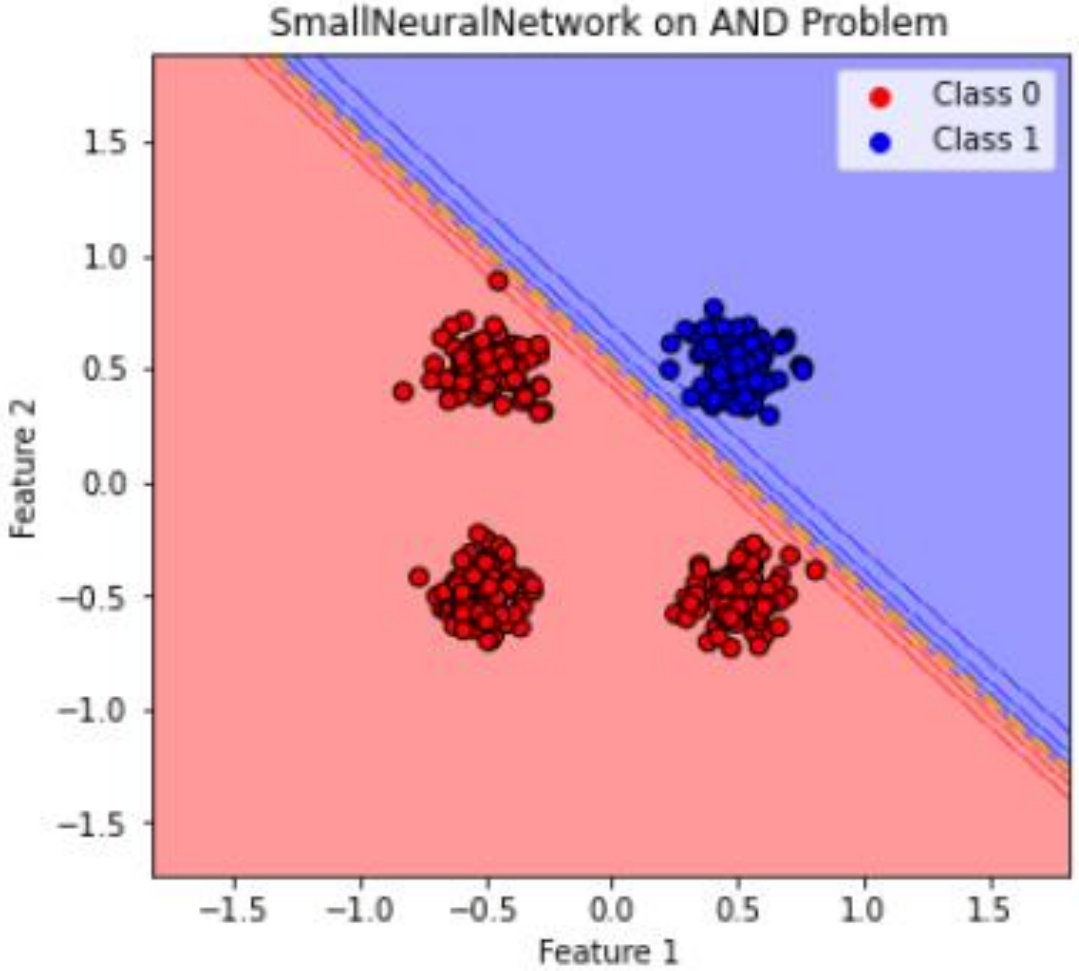
$$L(p) = p^{y} \cdot (1-p)^{(1-y)}$$

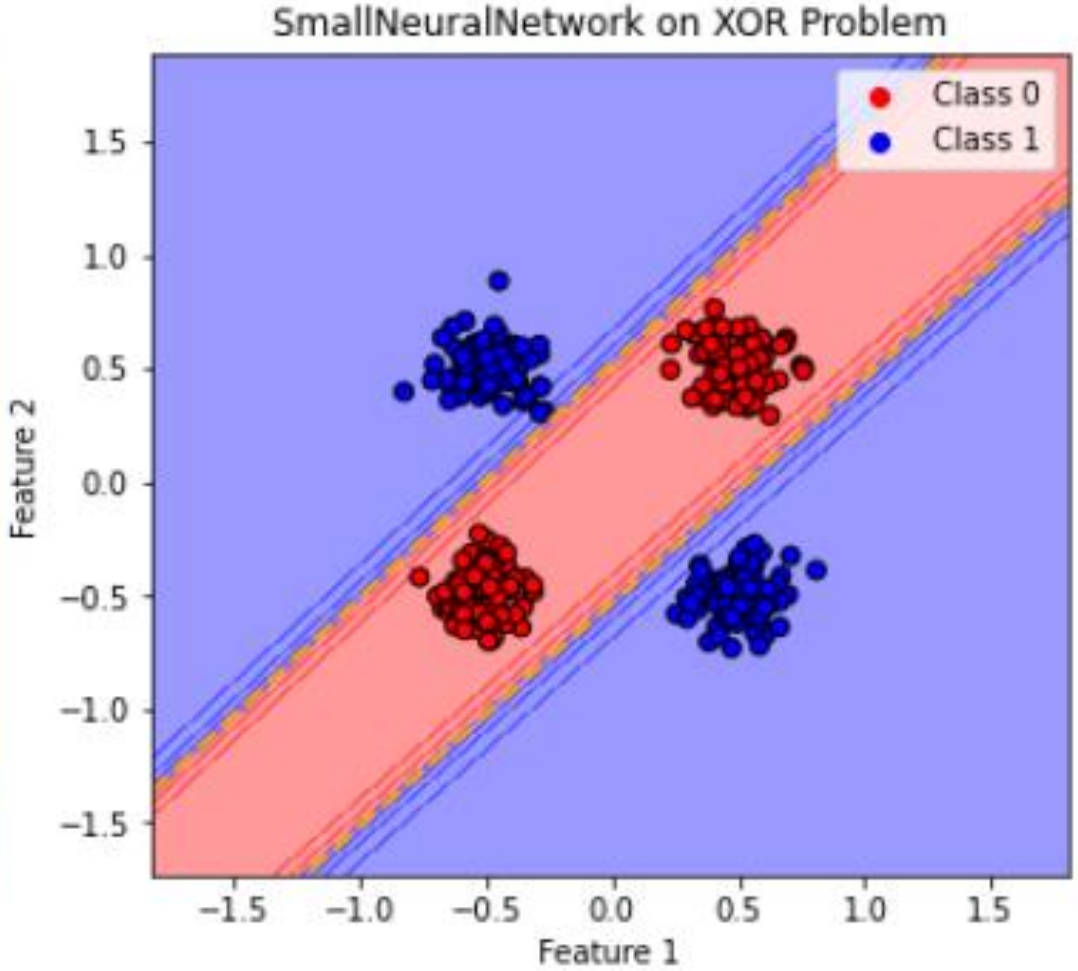
$$BCE = -[y \log(p) + (1-y) \log(1-p)]$$

Learning rate

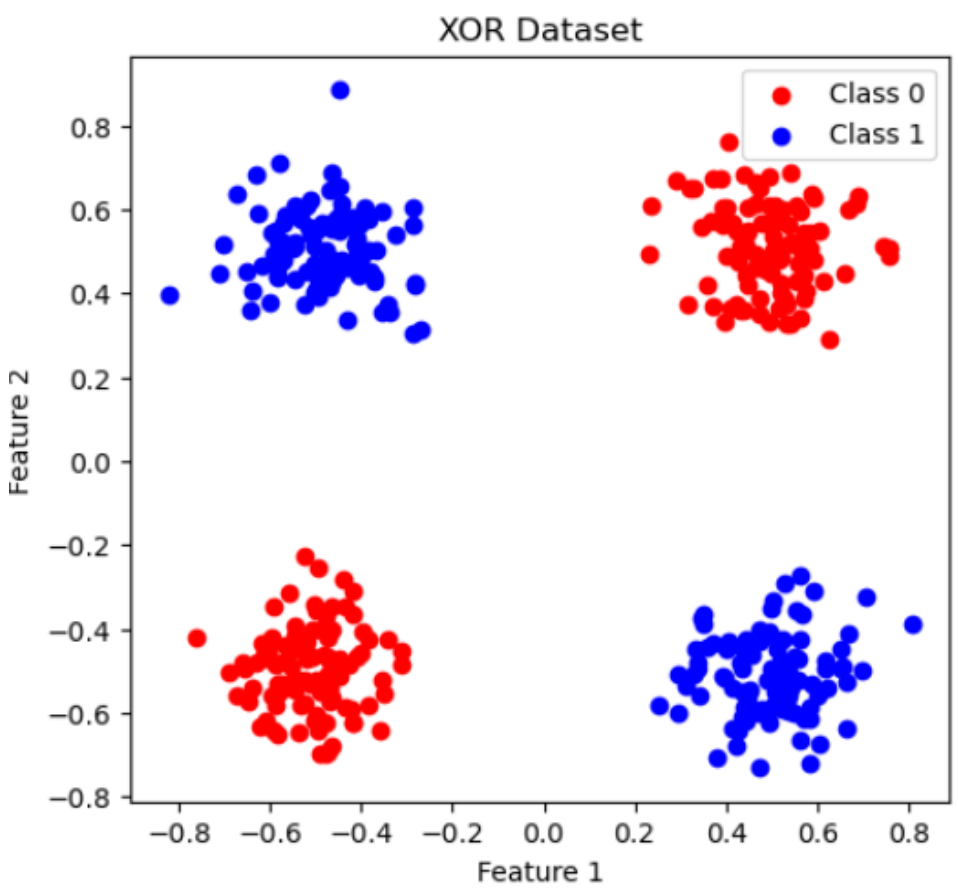
- Minimizing BCE → gradient descent → learning rate
- Effect of low value and high value for learning rate?

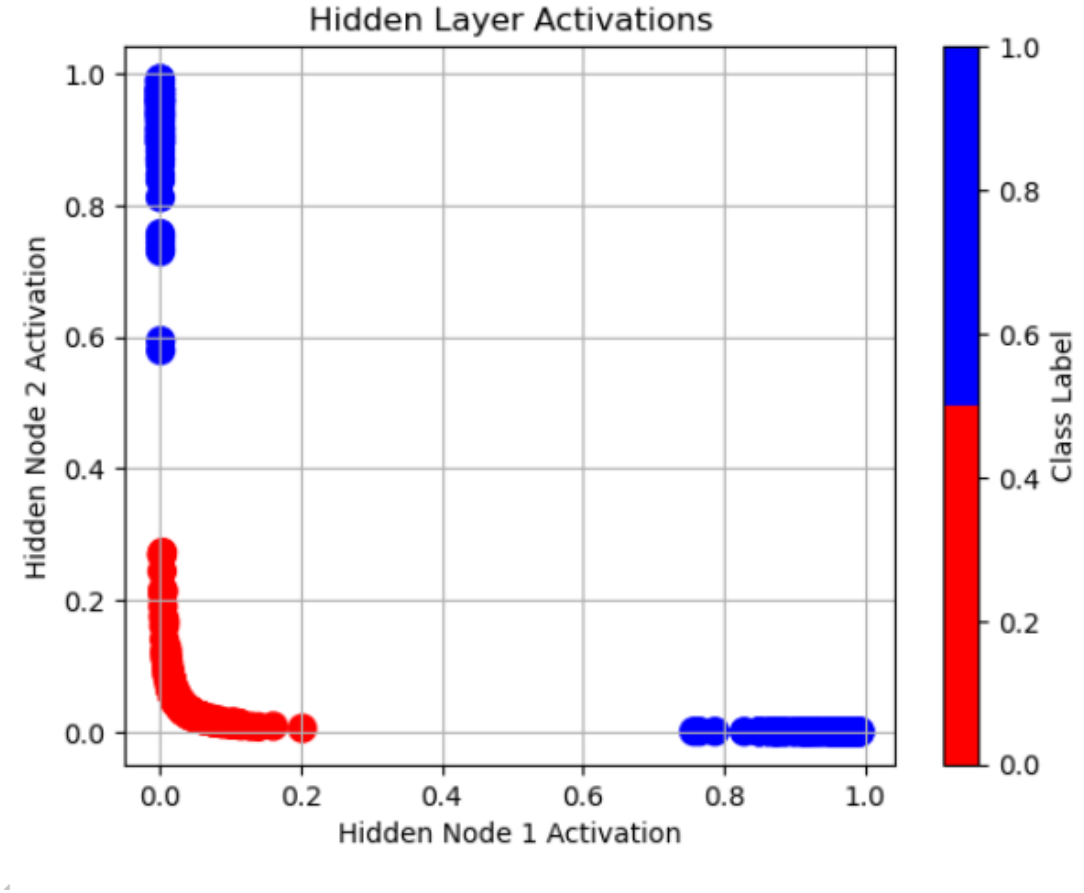
Why complexer models? Solve this problem:





Hidden layer / learned features of XOR





Hidden layers

- Q: How does the separability of the data in the hidden space compare to the original input space? What does this tell us about the role of the hidden layer?
- **A: Separation boundary is again linear!**
- **Hence, the role of the hidden layer is to transform the features in such a way, that the samples become linearly separable again with respect to the binary classes**

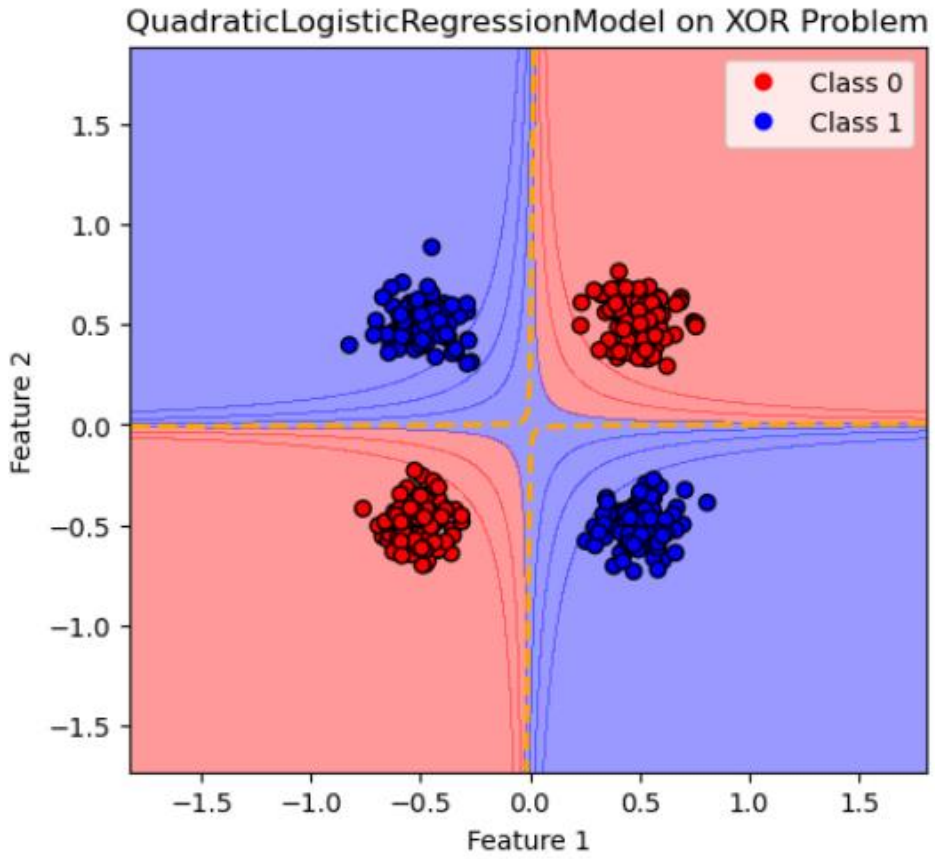
Hidden layers

- Q: Explain how the hidden layer activations enable the neural network to solve the XOR problem, which is not linearly separable in the original input space.
- **A: The (hidden layer) activation functions introduce non-linearities → classes become linearly separable again.**
- **Standard procedure of linear regression and minimizing a loss function can be employed again.**

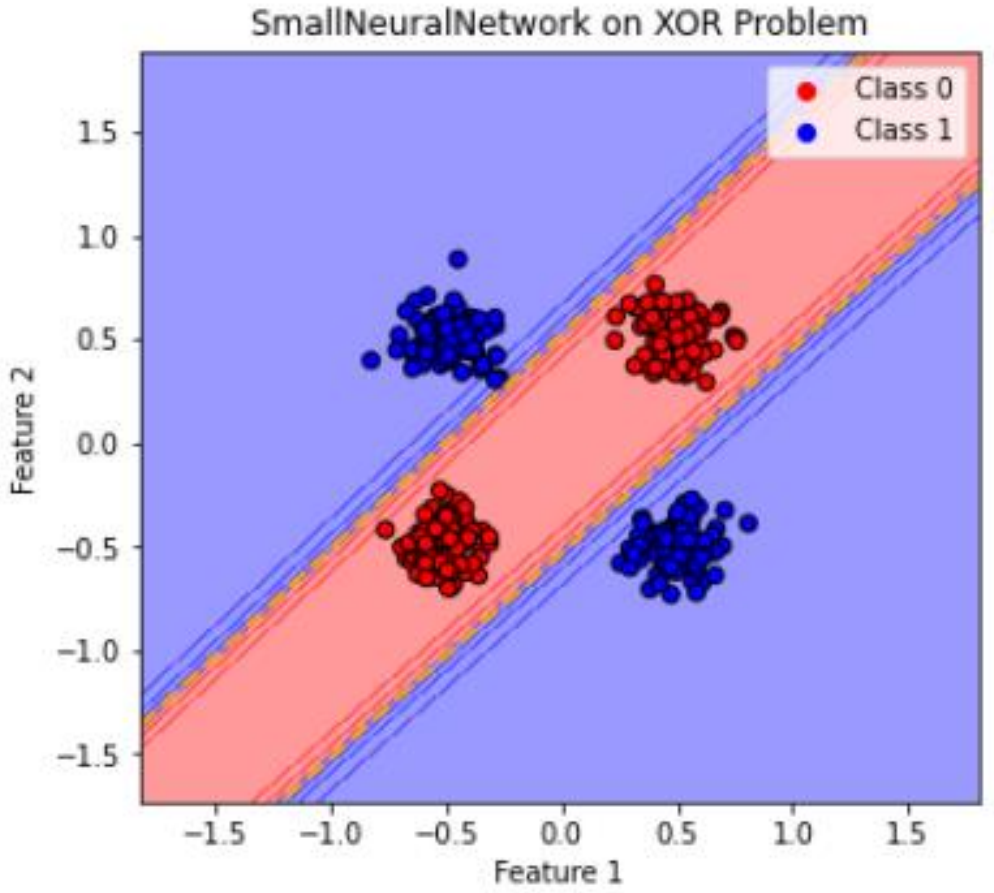
Manually engineering features vs. Learning features

Why is it beneficial to let the neural network learn feature transformations automatically rather than manually engineering features with logistic regression?

Answer: slides week 5



vs



neurons

- What might be the effect of changing the number of neurons in the hidden layer on the neural network's ability to solve the XOR problem? Try increasing or decreasing the number of hidden neurons and observe the impact.
- **The XOR problem is an example of a problem that cannot be solved using a simple linear regression or simple perceptron / single layer neural network because it is not linearly separable.**
- **A neural network with at least one layer can solve it.**
- **However, if we decrease the number of neurons in the hidden layer to fewer than 2, the network may struggle to learn the XOR problem effectively as it lacks the capacity to separate the data.**
- **Adding too many neurons might lead to the network learning to complex / specific patterns for that specific dataset … which is called…..**