Machine Learning in Finance - Homework 5 – Due 10.08.2024

Textbook reading:
Chapters 5 starting page 129

1. **for a range of inputs from -infinity to +infinity, what are the output ranges for the elu, exponential, gelu, linear, and softmax activation functions in https://www.tensorflow.org/api_docs/python/tf/keras/activations**

   - ReLU:

     $$\max(x, 0)$$
     $$\in (0,\infty)$$

   - Exponential Linear Unit (ELU):

     $$\alpha(e^x - 1), \qquad \text{if } x < 0$$
     $$x, \qquad \text{if } x \geq 0$$
     $$\in (-\alpha, \infty)$$

   - Gaussian Error Linear Unit (GELU):

     $$\frac{1}{2}x\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) = x\Phi(x)$$
     $$\in (-0.17, \infty)$$

   - Linear (pass-through):

     $$x$$
     $$\in (-\infty, \infty)$$

   - Softmax:

     $$\frac{e^{x_i}}{\Sigma e^{x_i}}$$
     $$\in (0,1)$$

     It also sums to one.

2. **What is regularization and why does it help create better models?**

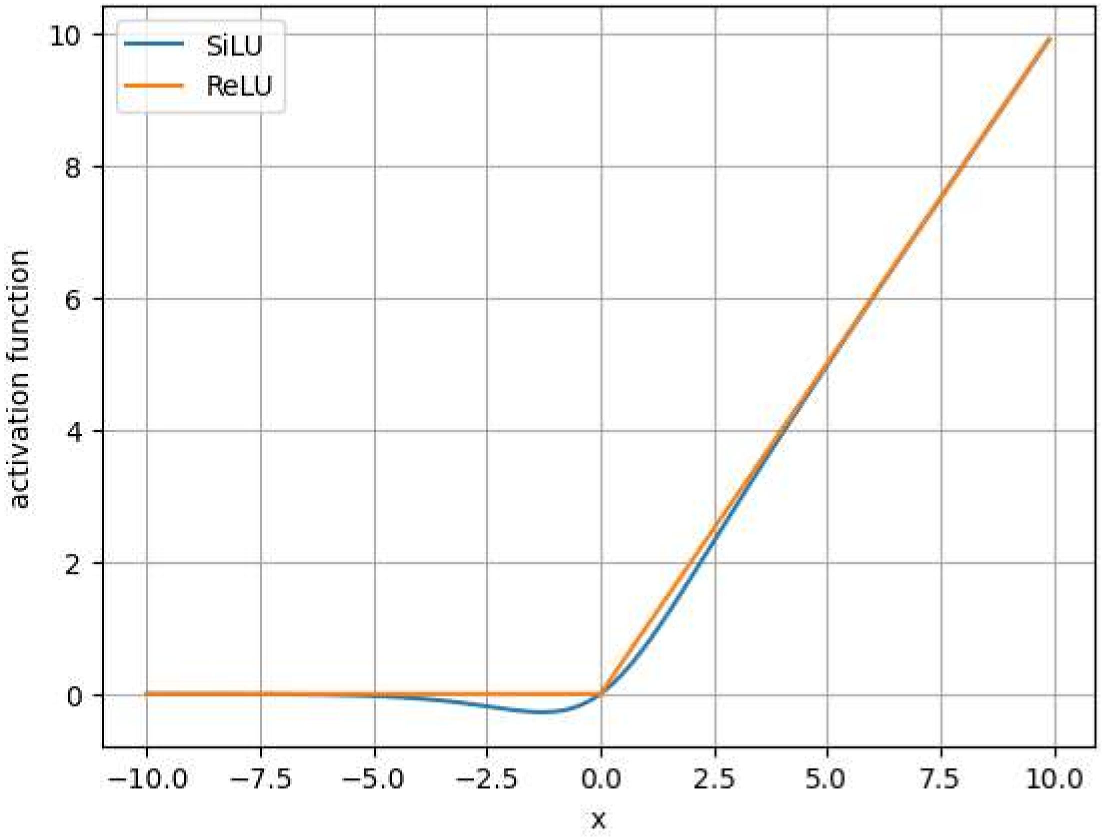
   Regularization adds a penalty term to the model's loss function, discouraging it from fitting the training data too closely by controlling the magnitude of model parameters.

   On the bias-variance trade off, regularization reduces variance by adding bias.

3. **Using matplotlib plot the swish function formula. How is it different from 'relu'? What advantages does it have over relu? What disadvantages does it have?**

   - Swish (SiLU):

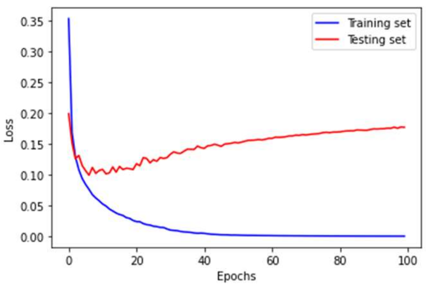
$$\frac{x}{1+e^{-x}}$$



ReLU has a kink at x=0. SiLU is differentiable everywhere, which has an advantage over ReLU in gradient calculations. On the other hand, the gradients of SiLU is slightly more computational costly.

4. **In Section5.2_ANN.ipynb, replace 'relu' with 'swish' and replace the single layer model with a model containing two hidden layers, each containing 45 neurons.**

   a. **Plot the loss history and the accuracy history. How do they compare to the same model using 'relu' instead of 'swish'?**



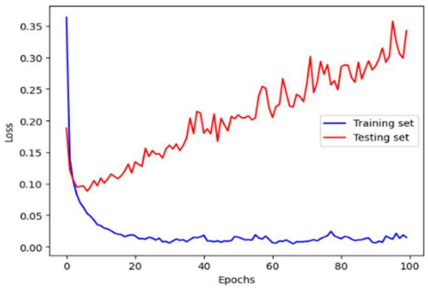


Figure 1. Loss history of ReLU(left) vs. SiLU(right)

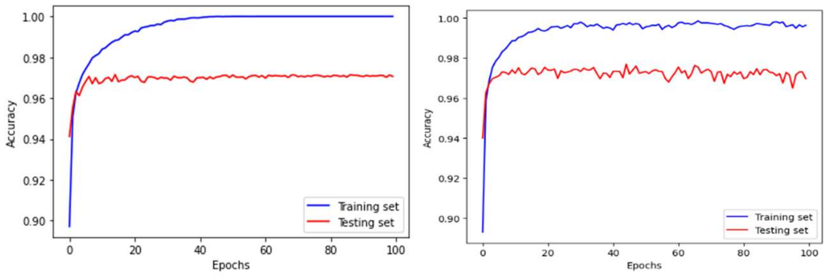


Figure 2. Accuracy history of ReLU(left) vs. SiLU(right)

By changing the model to 2 layers with SiLU activation function and 45 neurons each layer, the model seems to have overfitted the training dataset.

b. **How many parameters are estimated in this model. Use the formulas to compute the number of parameters to verify that the input values match the actual number of parameters.**

Input layer: 784

Hidden layer #1: 45

Hidden layer #2: 45

Output layer: 10

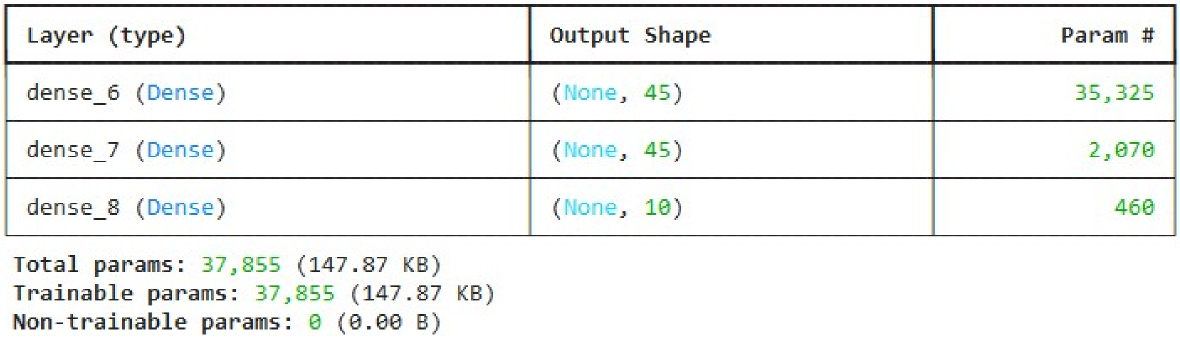
$$W_l = H_{l-1} \times H_l, \quad l = 1, 2, \ldots, L$$

$$B = H_1 + H_2 + \cdots + H_L$$

Total Parameters

= (784 x 45 + 45) + (45 x 45 + 45) + (45 x 10 + 10)

= 35325 + 2070 + 460 = 37855



Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 45)	35,325
dense_7 (Dense)	(None, 45)	2,070
dense_8 (Dense)	(None, 10)	460

```
Total params: 37,855 (147.87 KB)
Trainable params: 37,855 (147.87 KB)
Non-trainable params: 0 (0.00 B)
```

c. **Which layers in the model have no parameters to fit?**

Input layer.