Group Name:

Innovators Group:9

ML+CV Combined Project: Cell Segmentation

Group Members

Aadya Chinubhai AU2040232

Abhishu Oza AU2040027

Razin Karimi AU2040230

Nihar Jani AU2040205

Tasks Performed:

We are going to Implement:

1) Learnable Swish
2) Mexican Hat Functions
3) Sigmoid ReLu Shifted
4) Compare Different Activation Function

Learnable Swish

Code:

```
import tensorflow as tf

class LearnableSwish(tf.keras.layers.Layer):
    def __init__(self, **kwargs):
        super(LearnableSwish, self).__init__(**kwargs)

    def build(self, input_shape):
        self.beta = self.add_weight(shape=(1, ), initializer="ones", trainable=True)
        super(LearnableSwish, self).build(input_shape)

    def call(self, inputs):
        return tf.nn.swish(inputs * self.beta)
```

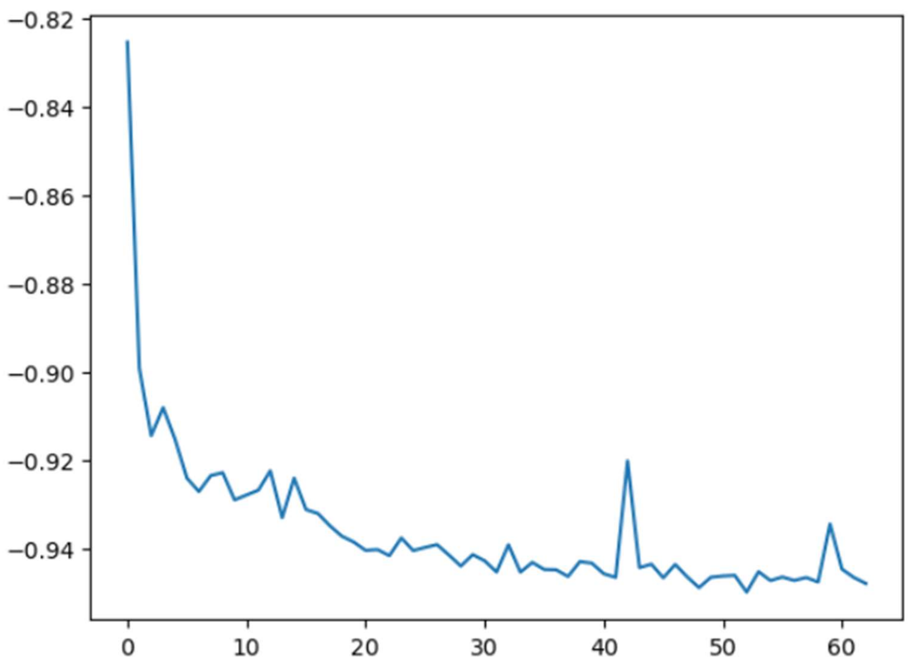
Training Part

```
cb = TimingCallback()

UNet_ls = UNet((96, 96, 3), activation_= LearnableSwish())
modells = UNet_ls.buildModel()
UNet_ls.CompileandSummarize(modells)
resultsls = modells.fit(x = X_train, y = Y_train, batch_size = 8, epochs=150, callbacks = [cb, tf.keras.callbacks.EarlyStopping(moni
tor='val_loss', patience=10)], validation_data = (X_test, y_test))

training_inferences['LearnableSwish'] = sum(cb.logs)
```

Result:

```
plt.plot(resultsls.history['val_loss'])
```

```
[<matplotlib.lines.Line2D at 0x7efdf10b9b50>]
```



Mexican ReLu

Code:

```
class MexicanReLU(tf.keras.layers.Layer):
    def __init__(self, alpha=1.0, sigma=1.0, mu=0.0):
        super(MexicanReLU, self).__init__()
        self.alpha = tf.Variable(alpha, trainable=True)
        self.sigma = tf.Variable(sigma, trainable=True)
        self.mu = tf.Variable(mu, trainable=True)

    def call(self, inputs):
        relu = tf.nn.relu(inputs)
        gaussian = self.alpha * tf.exp(-tf.square(inputs - self.mu) / (2 * tf.square(self.sigma))) * (inputs - self.mu)
        return relu + gaussian
```

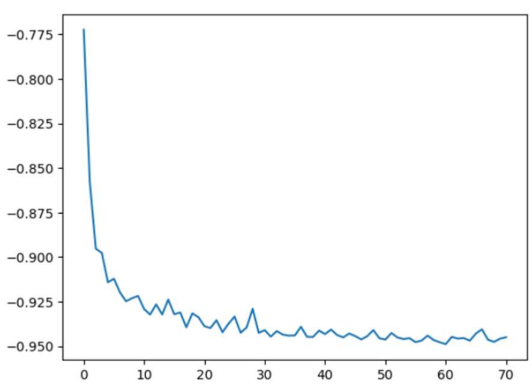
Training Part:

```
cb = TimingCallback()

UNet_1 = UNet((96, 96, 3), activation_= MexicanReLU())
model1 = UNet_1.buildModel()
UNet_1.CompileandSummarize(model1)
results1 = model1.fit(x = X_train, y = Y_train, batch_size = 8, epochs=150, callbacks = [cb, tf.keras.callbacks.EarlyStopping(monito
r='val_loss', patience=10)],  validation_data = (X_test, y_test))

training_inferences['MexicanReLU'] = sum(cb.logs)
```

Result:

```
plt.plot(results1.history['val_loss'])
```

```
[<matplotlib.lines.Line2D at 0x7f024876ef10>]
```



Compare Different Activation Function:

we will assess the relative strengths and weaknesses of six activation functions: Swish, ISRU, Trainable Leaky ReLU, Mish, Gated Swish, Learnable Swish, and Mexican Hat Function.

Swish, is a non-monotonic activation function defined as x multiplied by sigmoid(x). This function stands out for its smooth curve and ease of computation. Swish has demonstrated superior performance in image classification and language modeling applications.

ISRU is a smooth activation function that scales each neuron's output by the inverse square root of the sum of its squares. It has been shown to have comparable performance to ReLU in some tasks while avoiding overfitting and improving network generalization.

Trainable Leaky ReLU is a variant of traditional Leaky ReLU in which the negative slope can be learned during training. This activation function has been shown to enhance the performance of deep neural networks, particularly when the data distribution is unknown.

Mish is a non-monotonic activation function. It is defined as x multiplied by tanh(ln(1 + e^x)). Mish has been shown to outperform other activation functions in some tasks, such as image classification and object detection.

Gated Swish is a variant of Swish that introduces a gating mechanism. This mechanism enables the network to learn which parts of the input should be activated and which should be suppressed. Gated Swish has demonstrated superior performance in image classification and speech recognition tasks.

Learnable Swish is another variant of Swish that introduces learnable parameters, allowing the network to adapt the shape of the activation function to the specific task. This activation function has been shown to improve the performance of deep neural networks in some tasks, such as image classification and object detection.

.The Mexican hat function, which is defined as the second derivative of a Gaussian function, is used to detect edges and other features in images.

In conclusion, the selection of an appropriate activation function depends on the specific task at hand. Swish and its variants have demonstrated superior performance in image-related tasks, while ISRU and Mish have shown good generalization ability. Trainable Leaky ReLU can improve performance in deep neural networks, and the Mexican hat function is useful for detecting edges and features in images.