

ACTIVATION
FUNCTIONS



mish ACTIVATION



Swish Activation

Model	ResNet	WRN	DenseNet
LReLU	94.2	95.6	94.7
PReLU	94.1	95.1	94.5
Softplus	94.6	94.9	94.7
ELU	94.1	94.1	94.4
SELU	93.0	93.2	93.9
ReLU	93.8	95.3	94.8
Swish	94.7	95.5	94.8

Table 2: CIFAR-10 accuracy.

Model	ResNet	WRN	DenseNet
LReLU	74.2	78.0	83.3
PReLU	74.5	77.3	81.5
Softplus	76.0	78.4	83.7
ELU	75.0	76.0	80.6
SELU	73.2	74.3	80.8
ReLU	74.2	77.8	83.7
Swish	75.1	78.5	83.8

Table 3: CIFAR-100 accuracy.

Mish Activation

Table 6. CIFAR-100 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-110	74.41%	74.13%	73%
WRN 22-10	72.32%	71.89%	72.2%
WRN 40-4	69.52%	69.59%	69.35%
DenseNet - 121	66.31%	65.91%	65.50%
DenseNet - 169	65.38%	65.69%	64.99%
ResNext-50	67.58%	66.72%	67.52%
MobileNet v1	50.09%	49.95%	49.20%
SE Net-18	64.38%	63.89%	62.71%
Shuffle Net v2	59.35%	58.91%	58.56%
Squeeze Net	63.07%	62.11%	60.92%

Mish Activation

Table 6. CIFAR-100 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-110	74.41%	74.13%	73%
WRN 22-10	72.32%	71.89%	72.2%
WRN 40-4	69.52%	69.59%	69.35%
DenseNet - 121	66.31%	65.91%	65.50%
DenseNet - 169	65.38%	65.69%	64.99%
ResNext-50	67.58%	66.72%	67.52%
MobileNet v1	50.09%	49.95%	49.20%
SE Net-18	64.38%	63.89%	62.71%
Shuffle Net v2	59.35%	58.91%	58.56%
Squeeze Net	63.07%	62.11%	60.92%

Mish Activation

$$f(x) = x \cdot \text{sigmoid}(x)$$

.

Mish Activation

$$f(x) = x \cdot \text{sigmoid}(x)$$



$$f(x) = x \cdot \tanh(x)$$

$$f(x) = x \cdot \tanh\left(\frac{\ln(1 + e^x)}{x}\right)$$



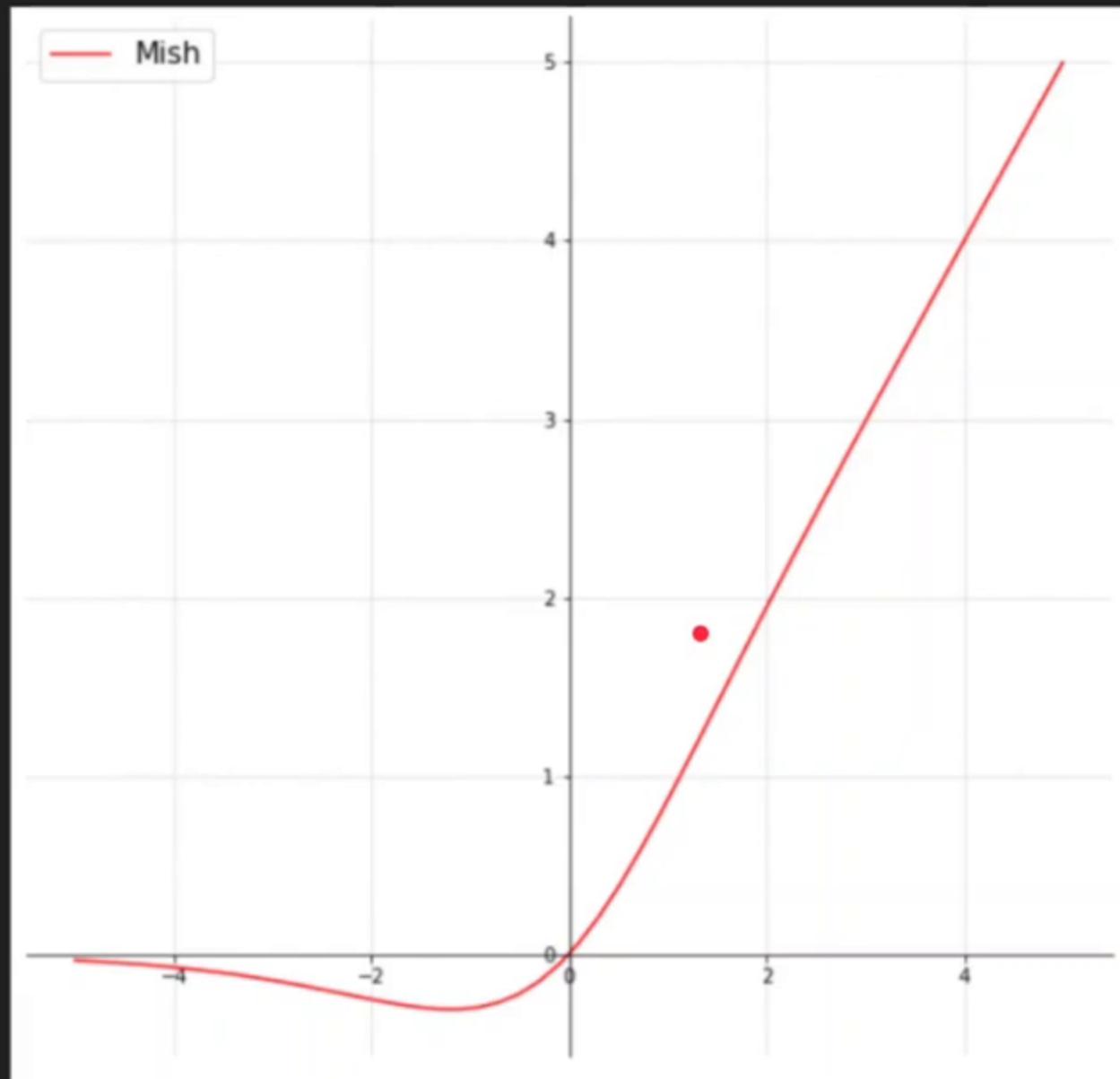
Mish Activation

$$f(x) = x \cdot \text{sigmoid}(x)$$

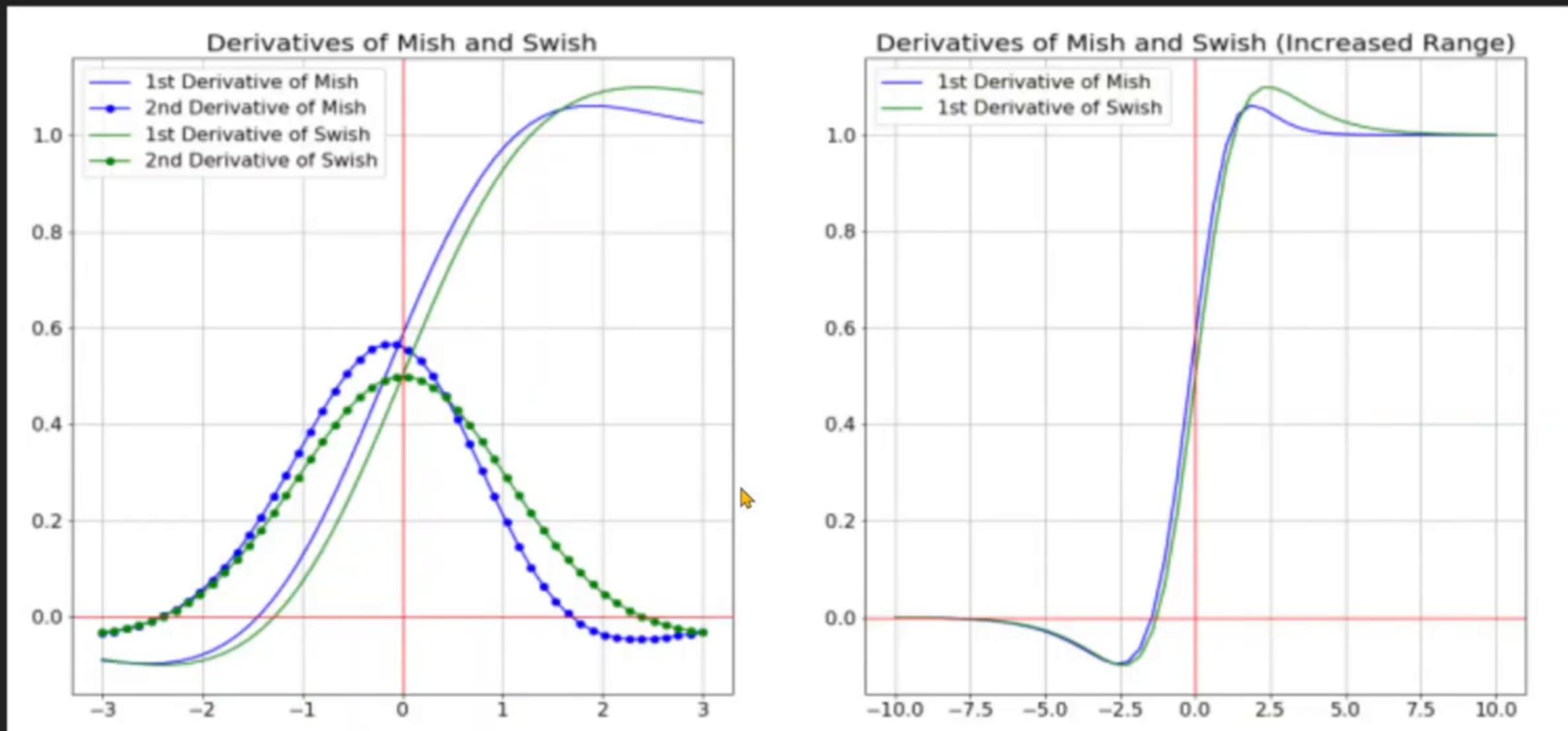


$$f(x) = x \cdot \tanh(x)$$

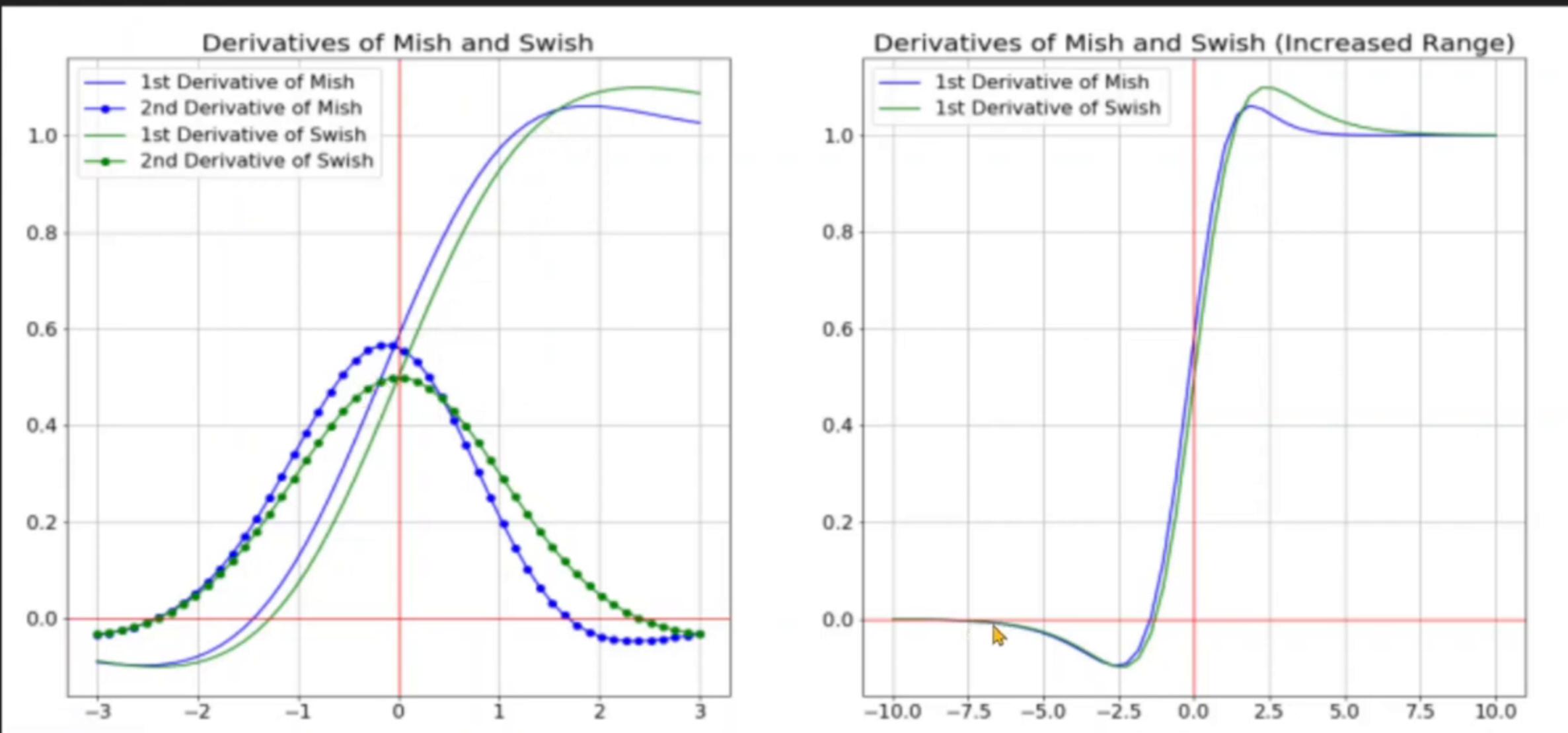
$$f(x) = x \cdot \tanh \left(\underline{\ln(1 + e^x)} \right)$$



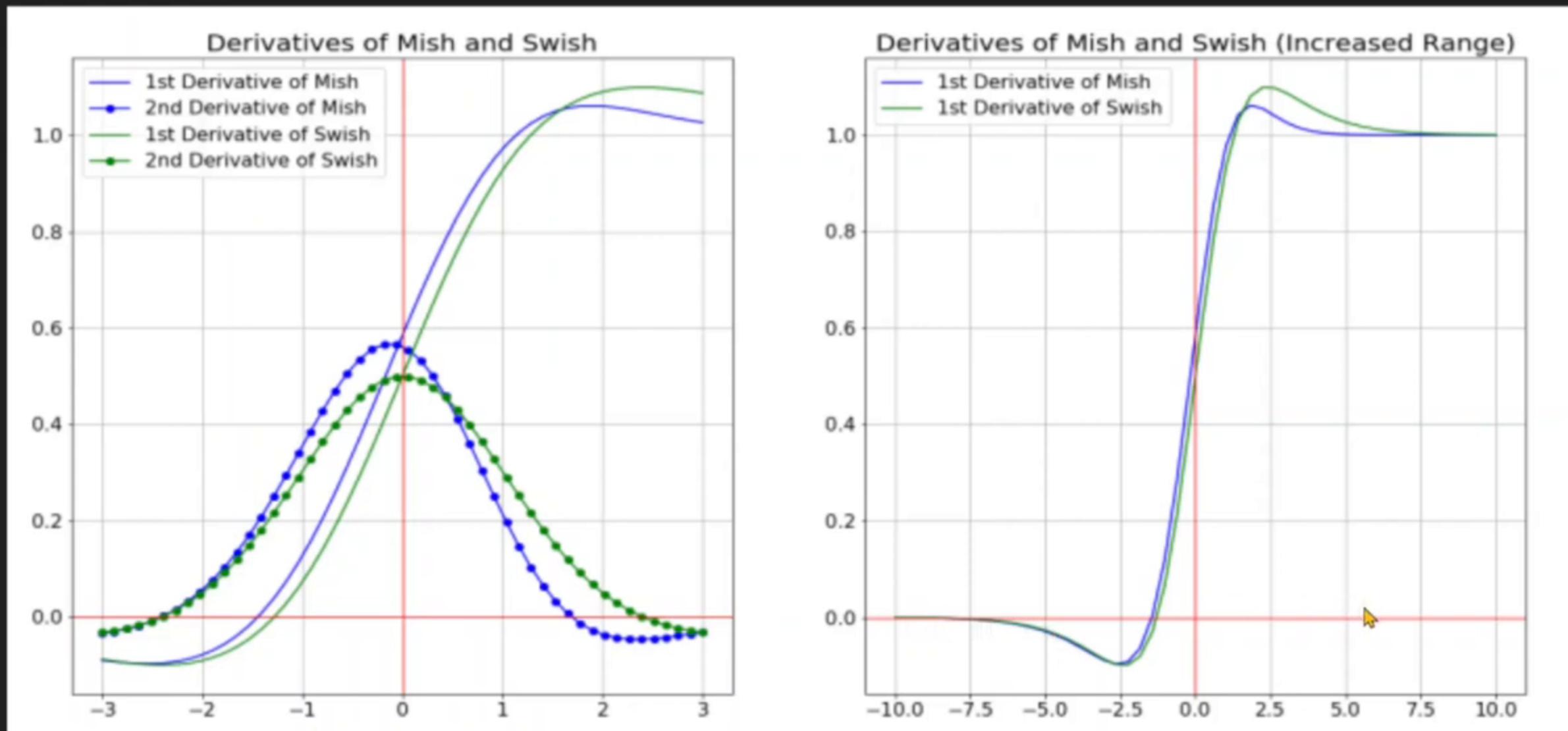
Derivative of Mish



Derivative of Mish

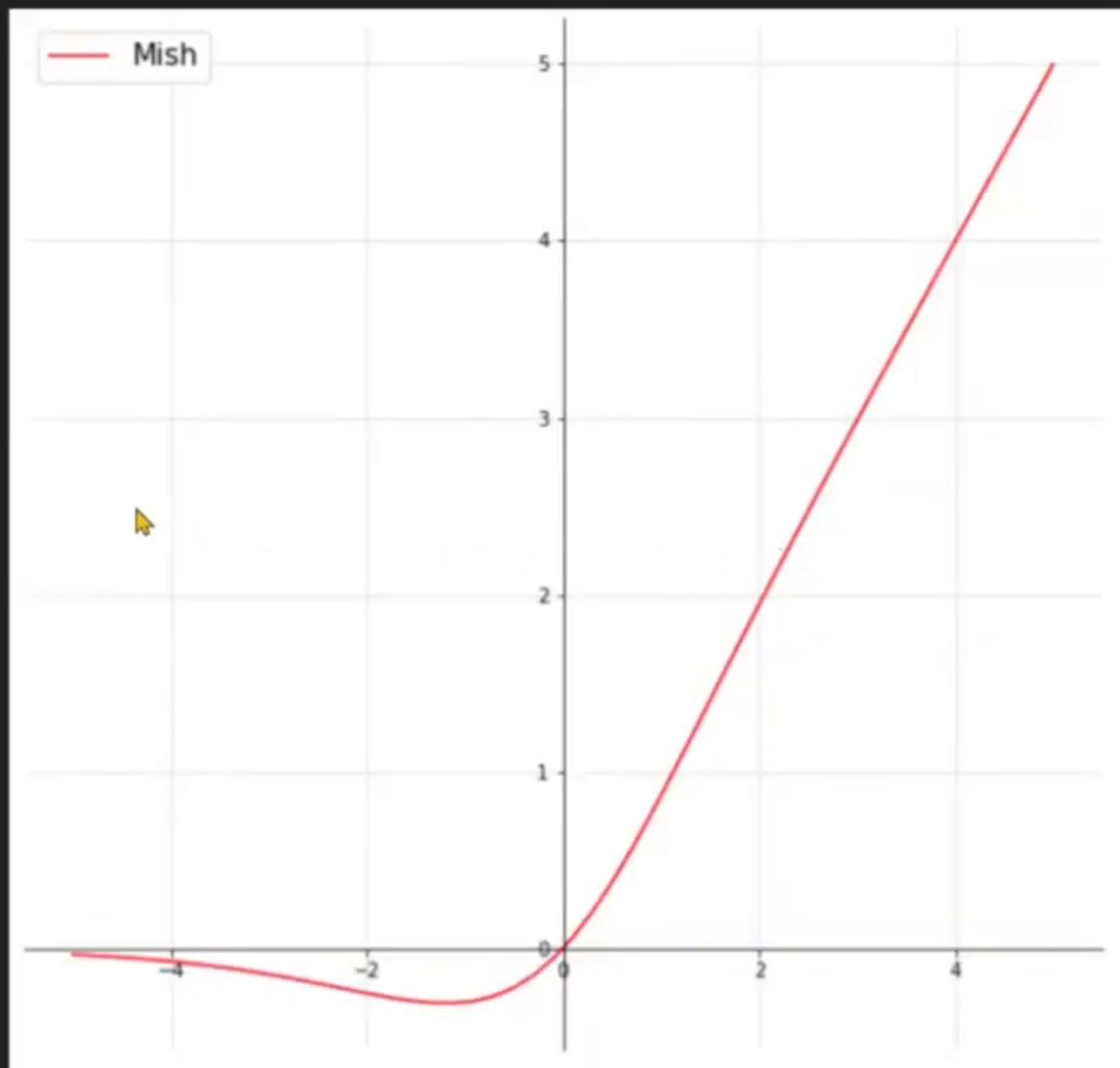


Derivative of Mish



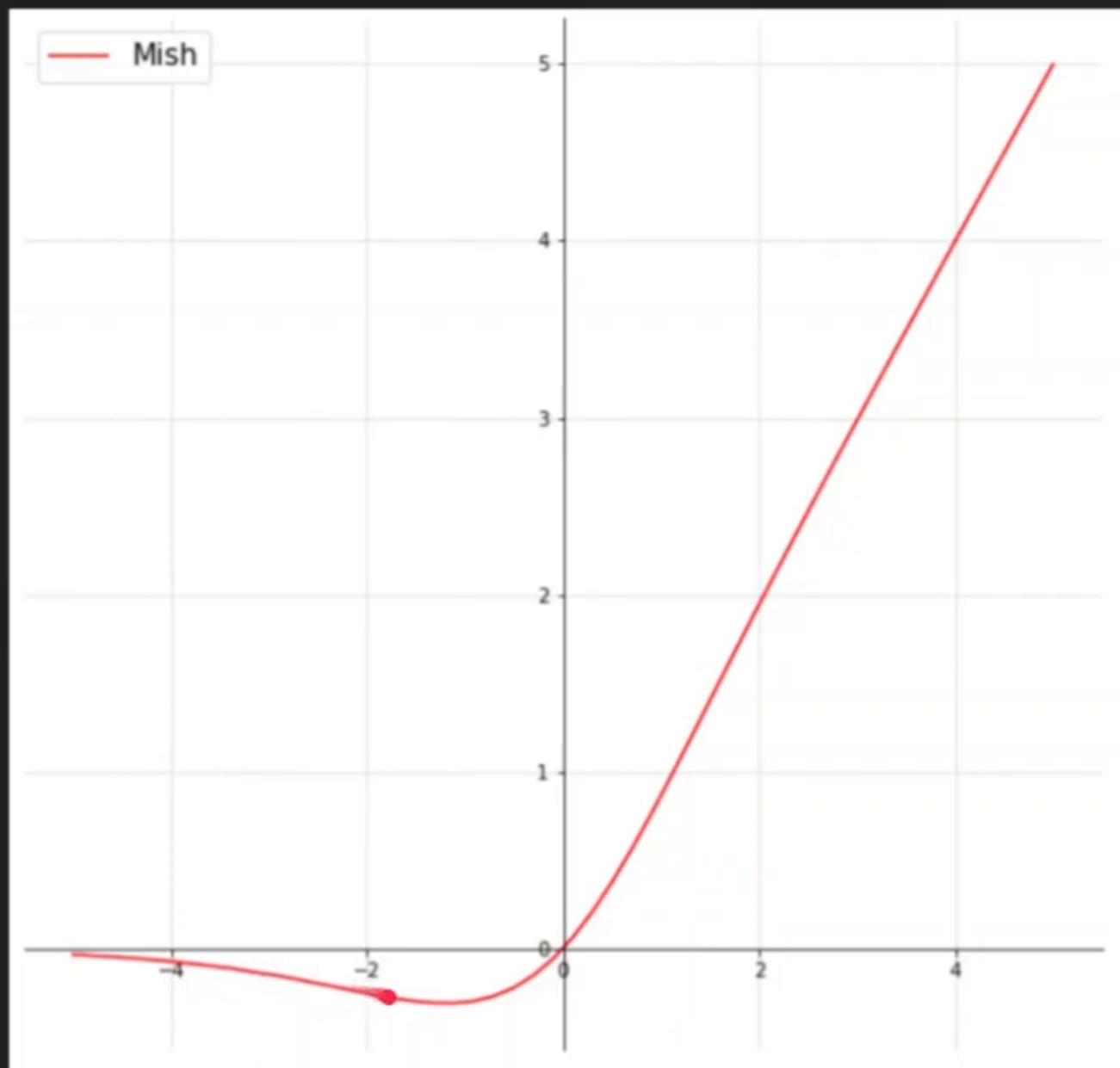
Properties

- Continuous
- Non-monotonicity
- Zero-centered
- Unbounded



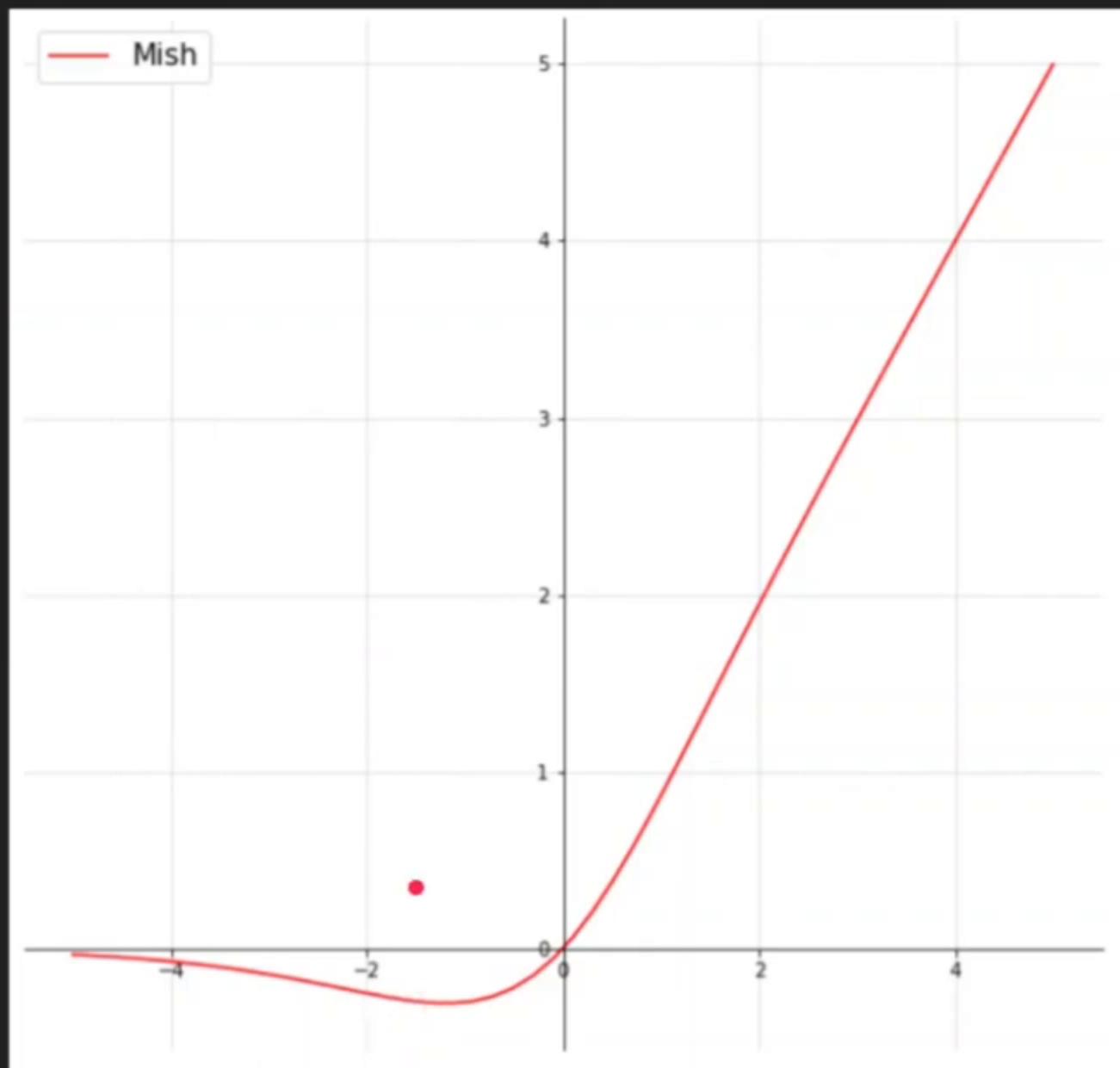
Properties

- Continuous
- Non-monotonicity
- Zero-centered
- Unbounded



Properties

- Continuous
- Non-monotonicity
- Zero-centered
- Unbounded



Performance

Table 5. CIFAR-10 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-20	92.02%	91.61%	91.71%
WRN 10-2	86.83%	86.56%	84.56%
SimpleNet	91.70%	91.44%	91.16%
Xception Net	88.73%	88.56%	88.38%
Capsule Net	83.15%	82.48%	82.19%
Inception ResNet v2	85.21%	84.96%	82.22%

•

Performance

Table 5. CIFAR-10 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-20	92.02%	91.61%	91.71%
WRN 10-2	86.83%	86.56%	84.56%
SimpleNet	91.70%	91.44%	91.16%
Xception Net	88.73%	88.56%	88.38%
Capsule Net	83.15%	82.48%	82.19%
Inception ResNet v2	85.21%	84.96%	82.22%

Performance

Table 5. CIFAR-10 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-20	92.02%	91.61%	91.71%
WRN 10-2	86.83%	86.56%	84.56%
SimpleNet	91.70%	91.44%	91.16%
Xception Net	88.73%	88.56%	88.38%
Capsule Net	83.15%	82.48%	82.19%
Inception ResNet v2	85.21%	84.96%	82.22%

Performance

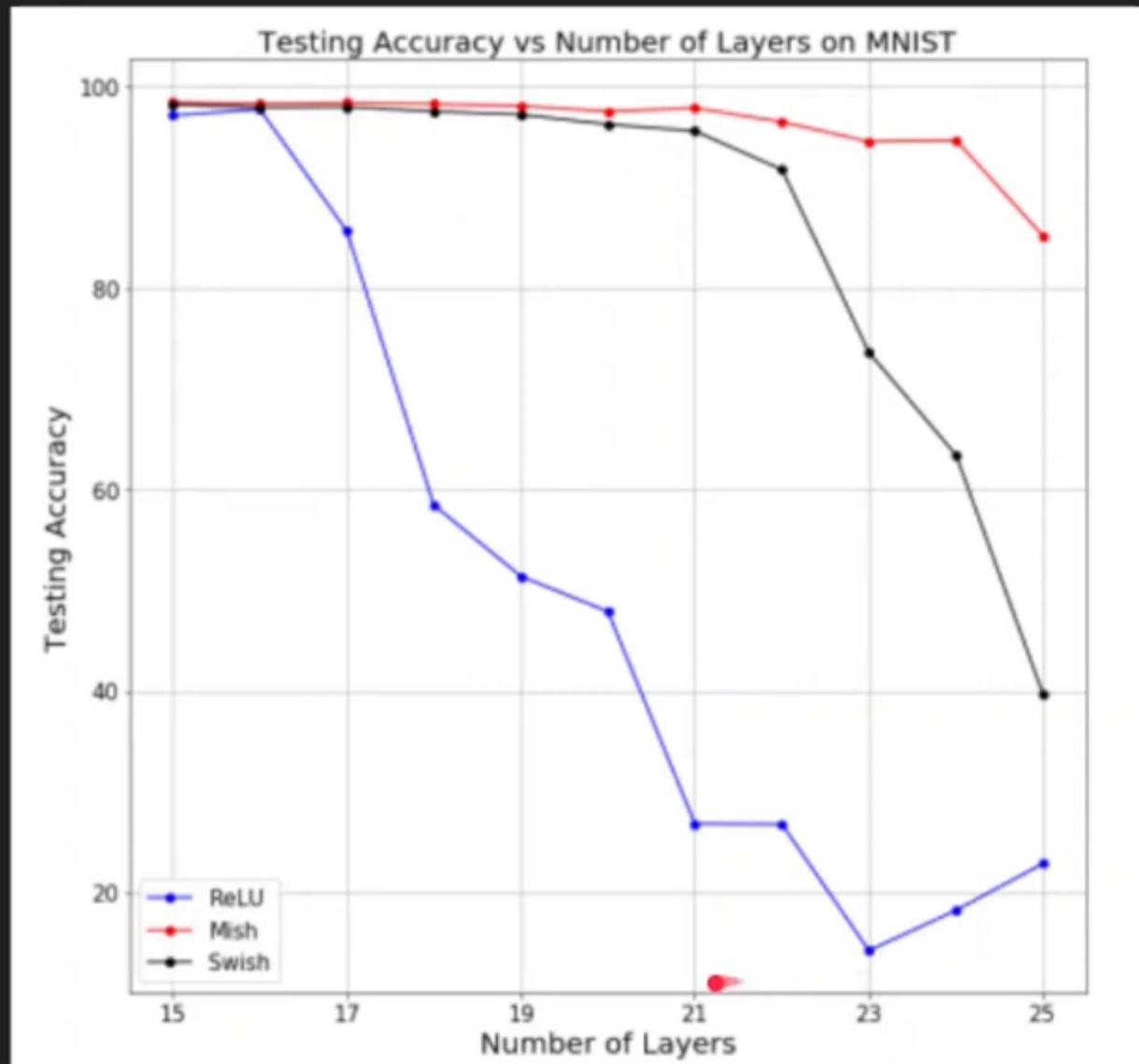
Table 6. CIFAR-100 Results (Test Top-1 Accuracy)

Model	Mish	Swish	ReLU
ResNet v2-110	74.41%	74.13%	73%
WRN 22-10	72.32%	71.89%	72.2%
WRN 40-4	69.52%	69.59%	69.35%
DenseNet - 121	66.31%	65.91%	65.50%
DenseNet - 169	65.38%	65.69%	64.99%
ResNext-50	67.58%	66.72%	67.52%
MobileNet v1	50.09%	49.95%	49.20%
SE Net-18	64.38%	63.89%	62.71%
Shuffle Net v2	59.35%	58.91%	58.56%
Squeeze Net	63.07%	62.11%	60.92%

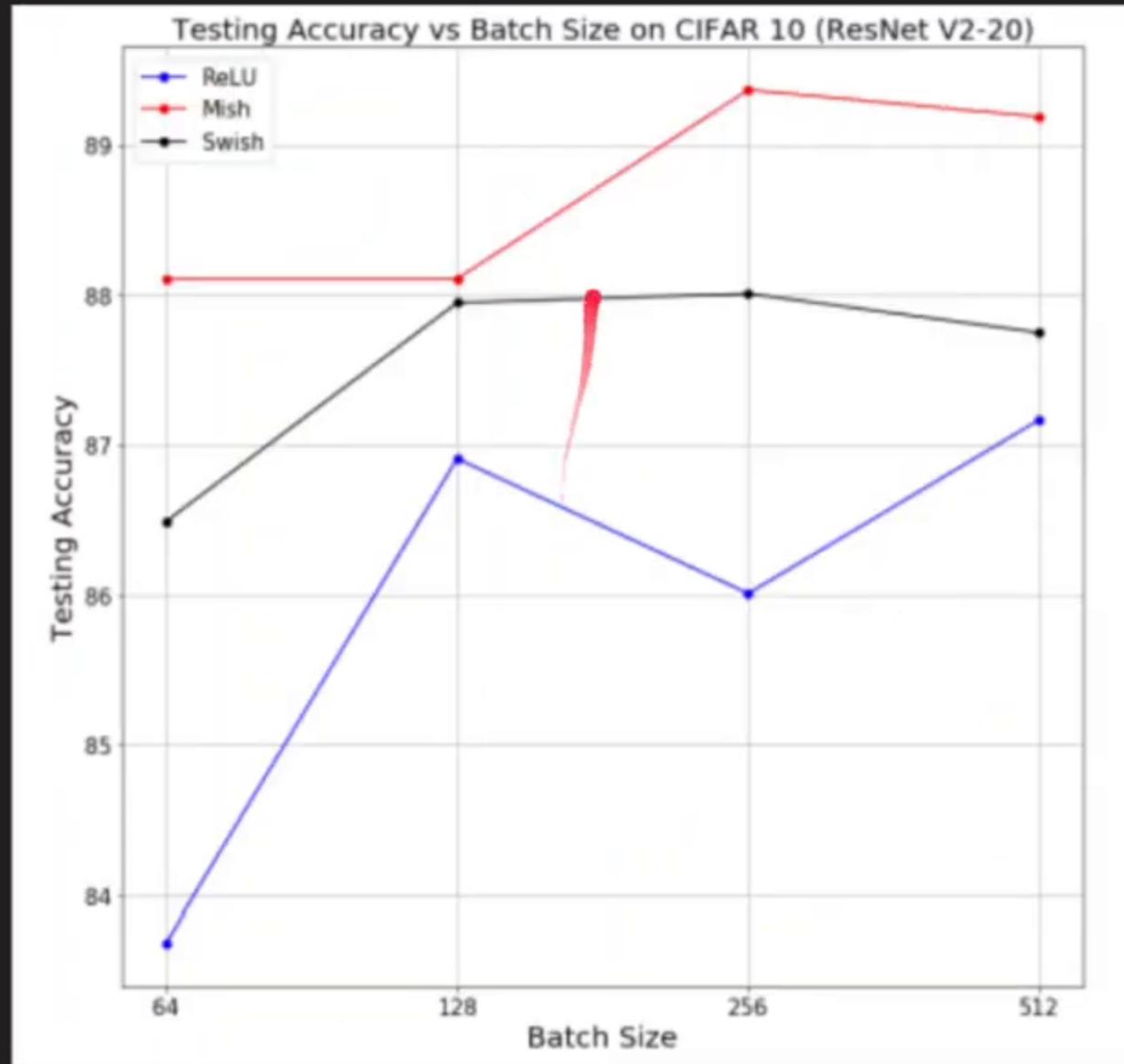
Ablation Studies

Ablation Studies

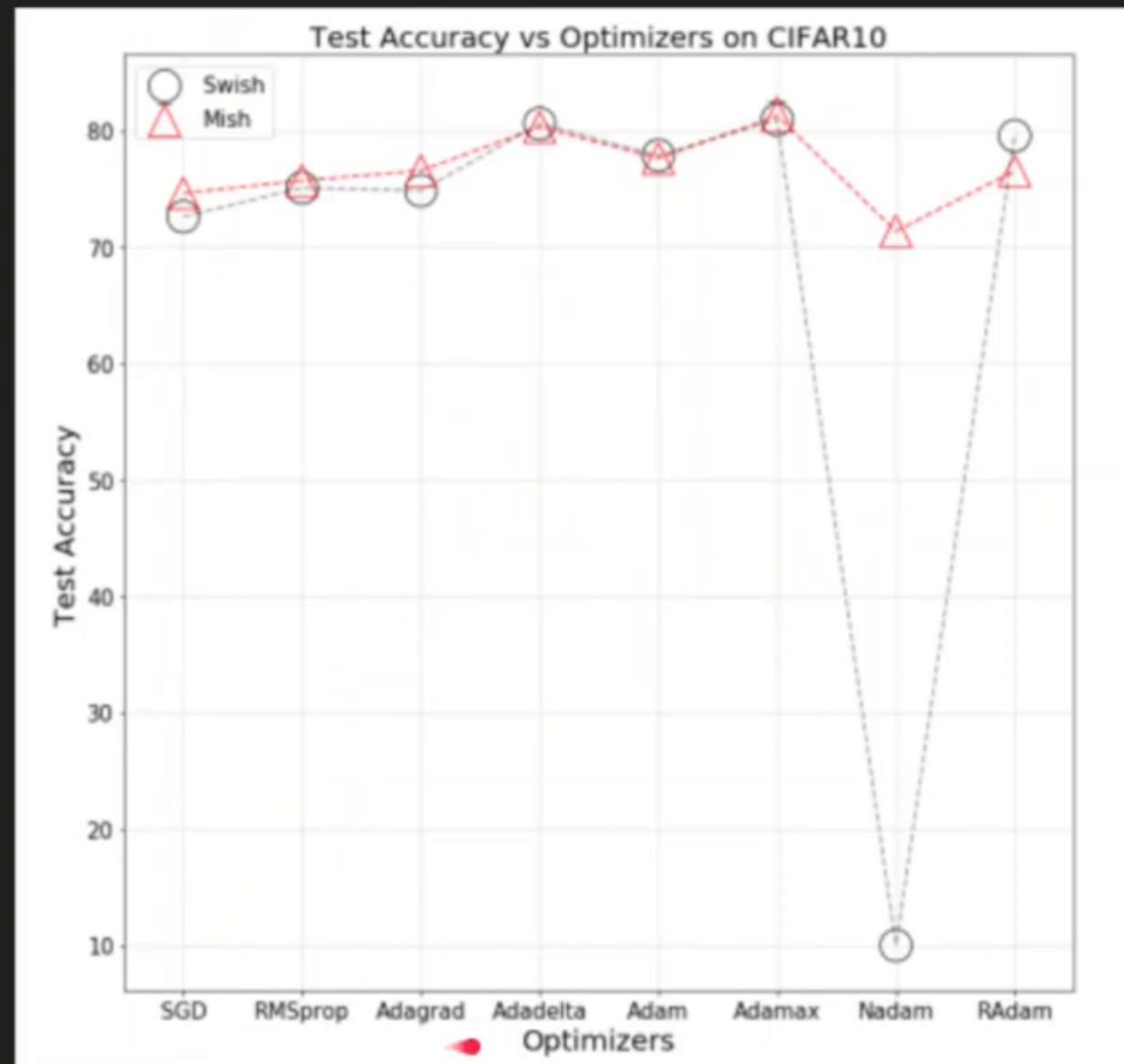
Ablations - Number of Layers



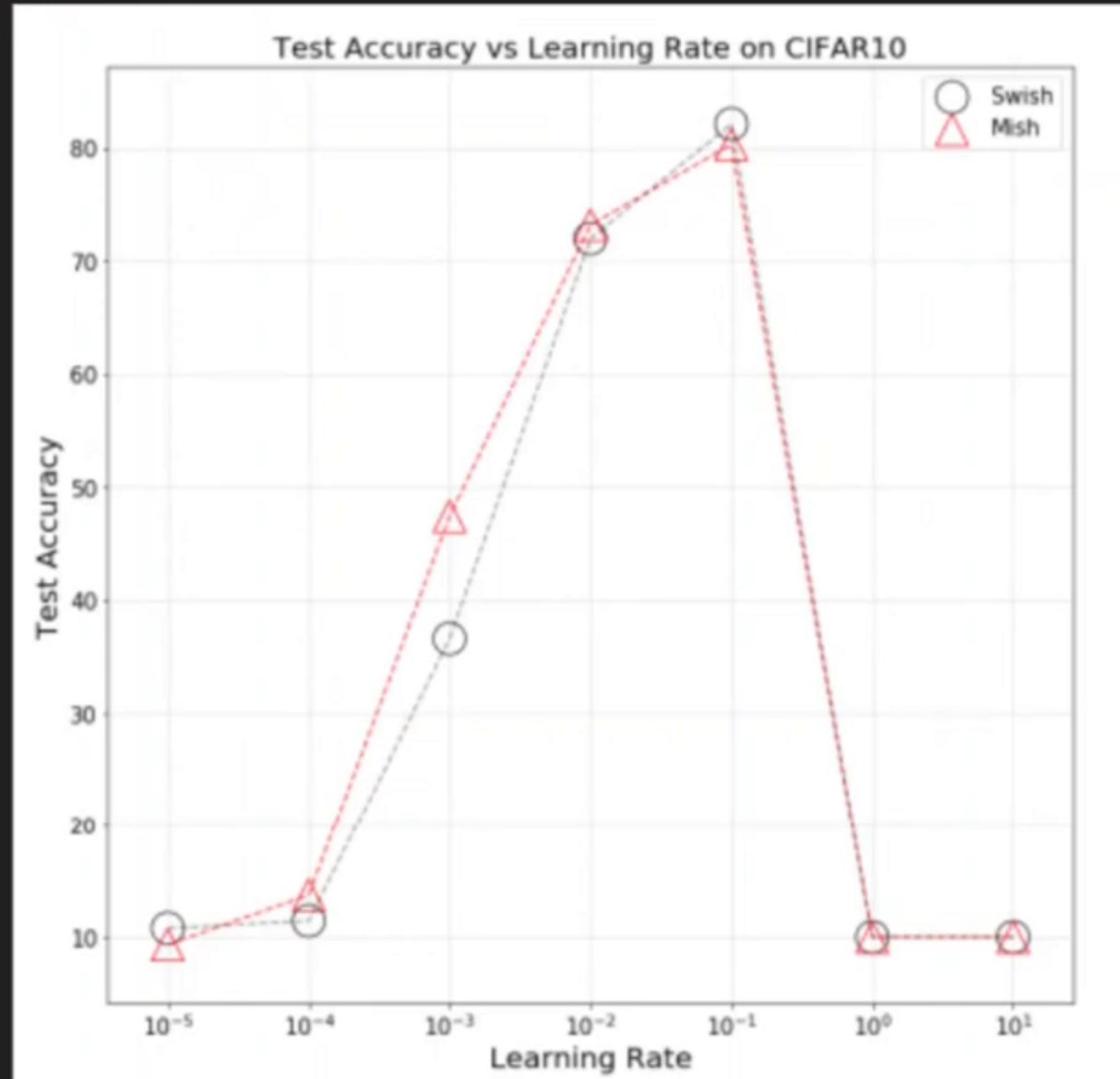
Ablations - Batch Size



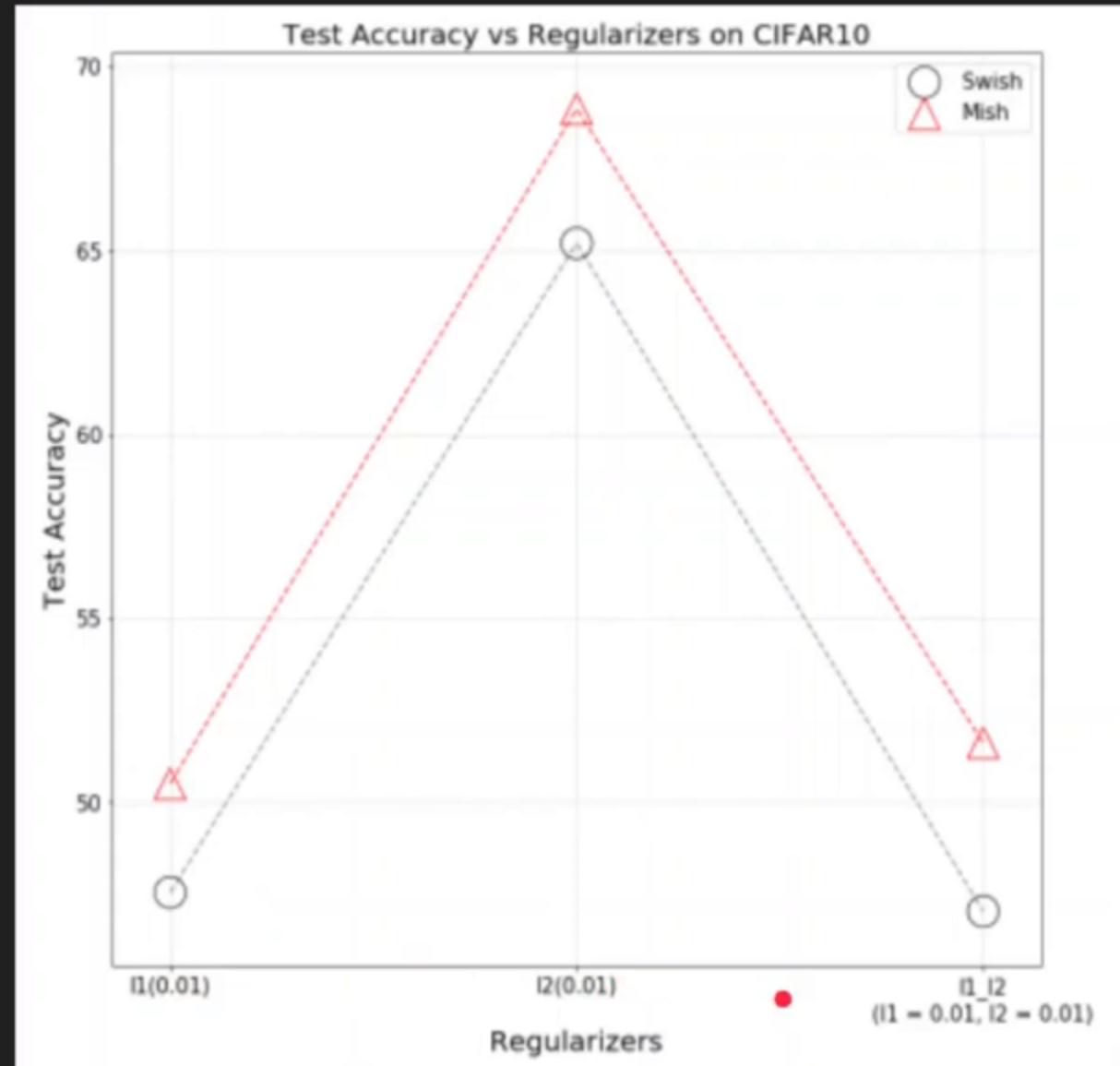
Ablations - Optimizers



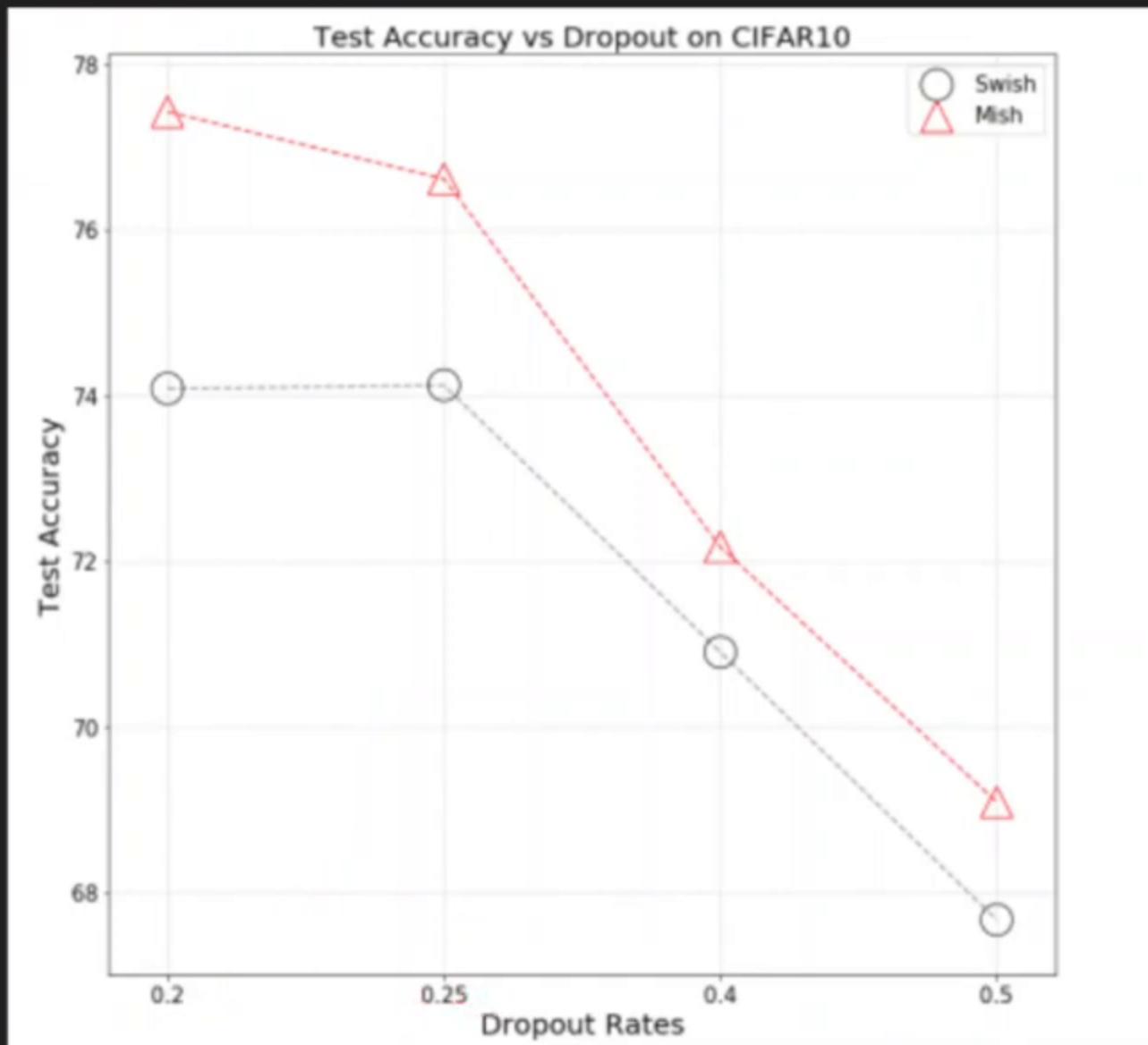
Ablations - Learning rates



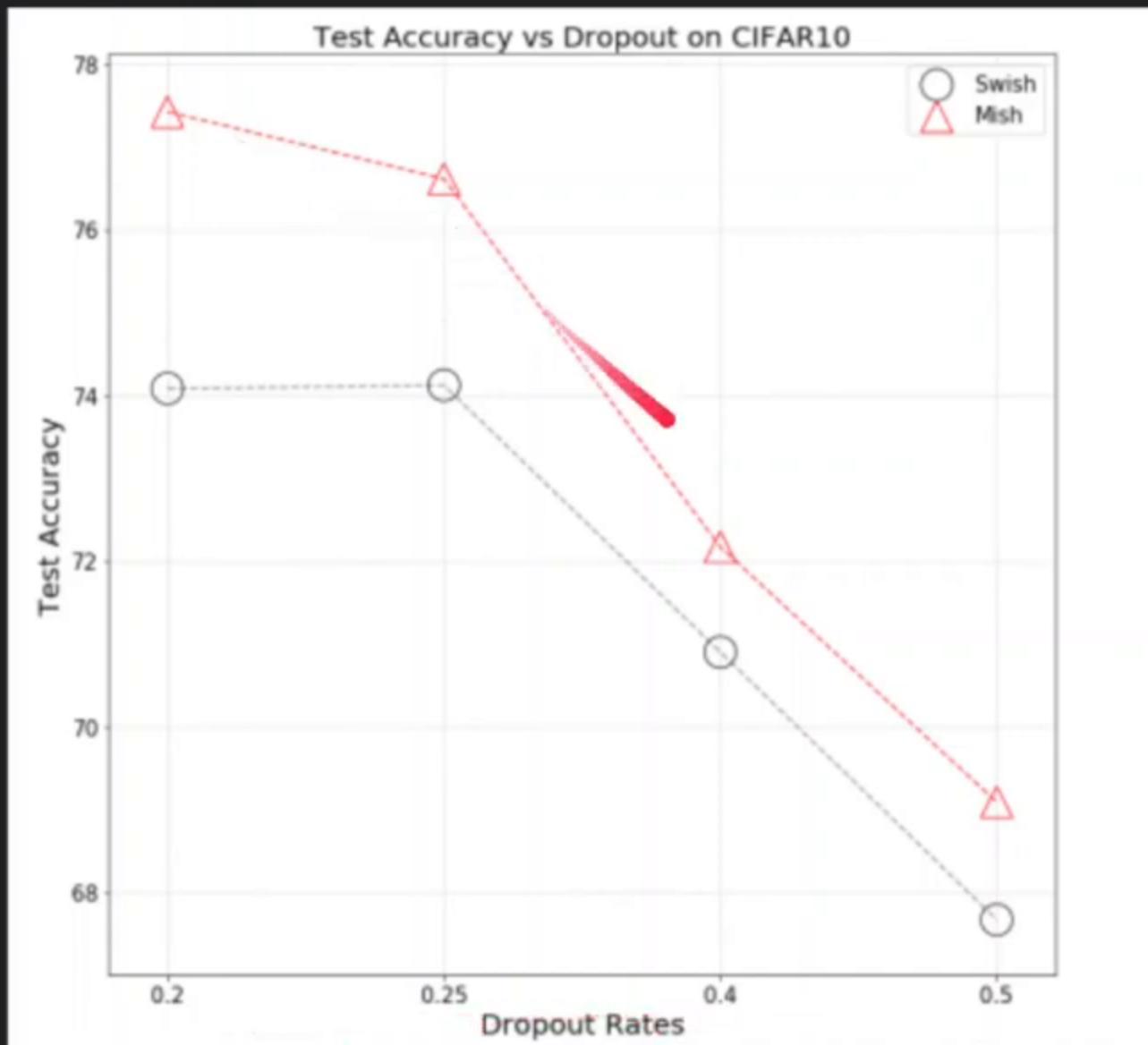
Ablations - Regularizers



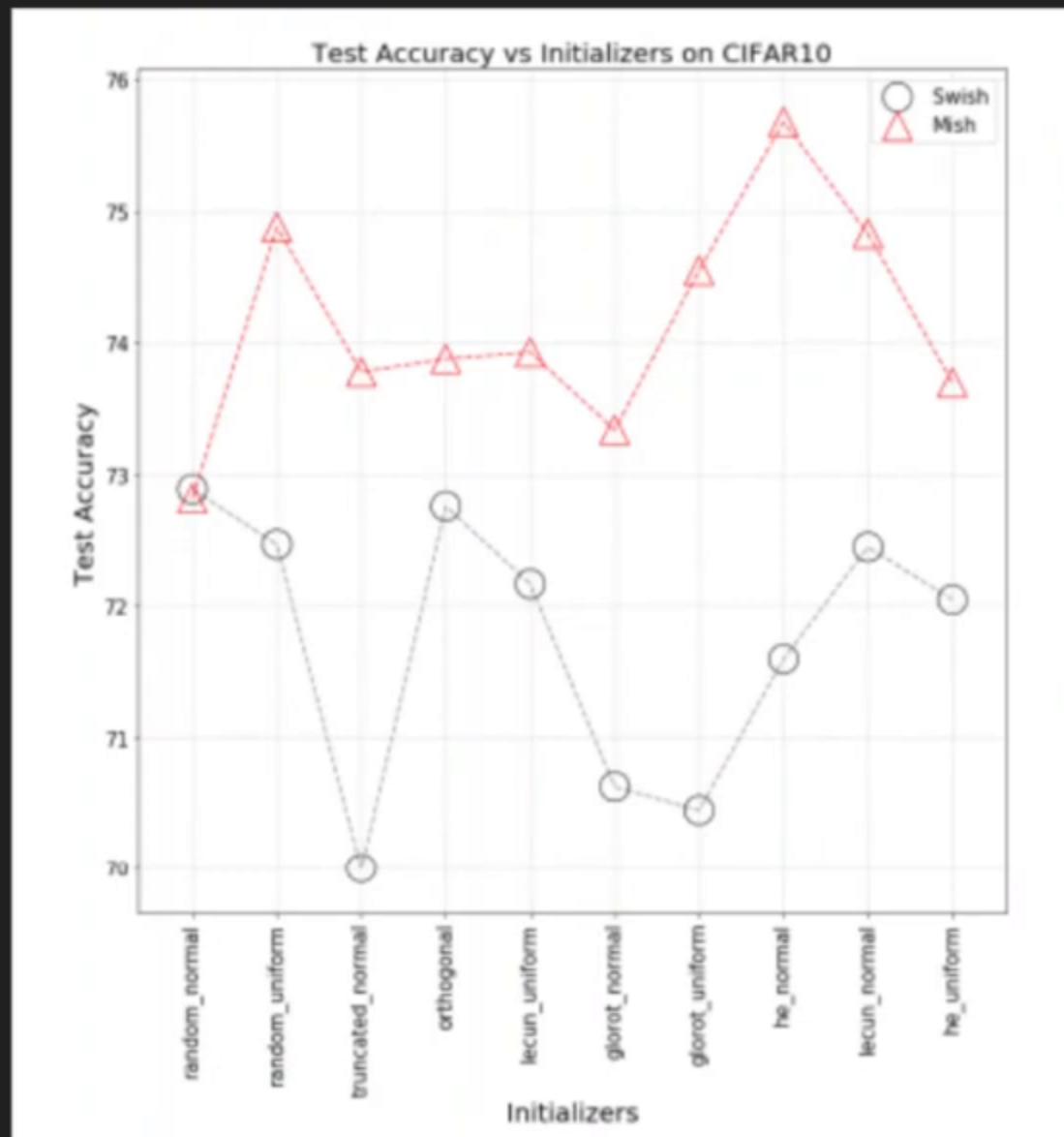
Ablations - Dropout



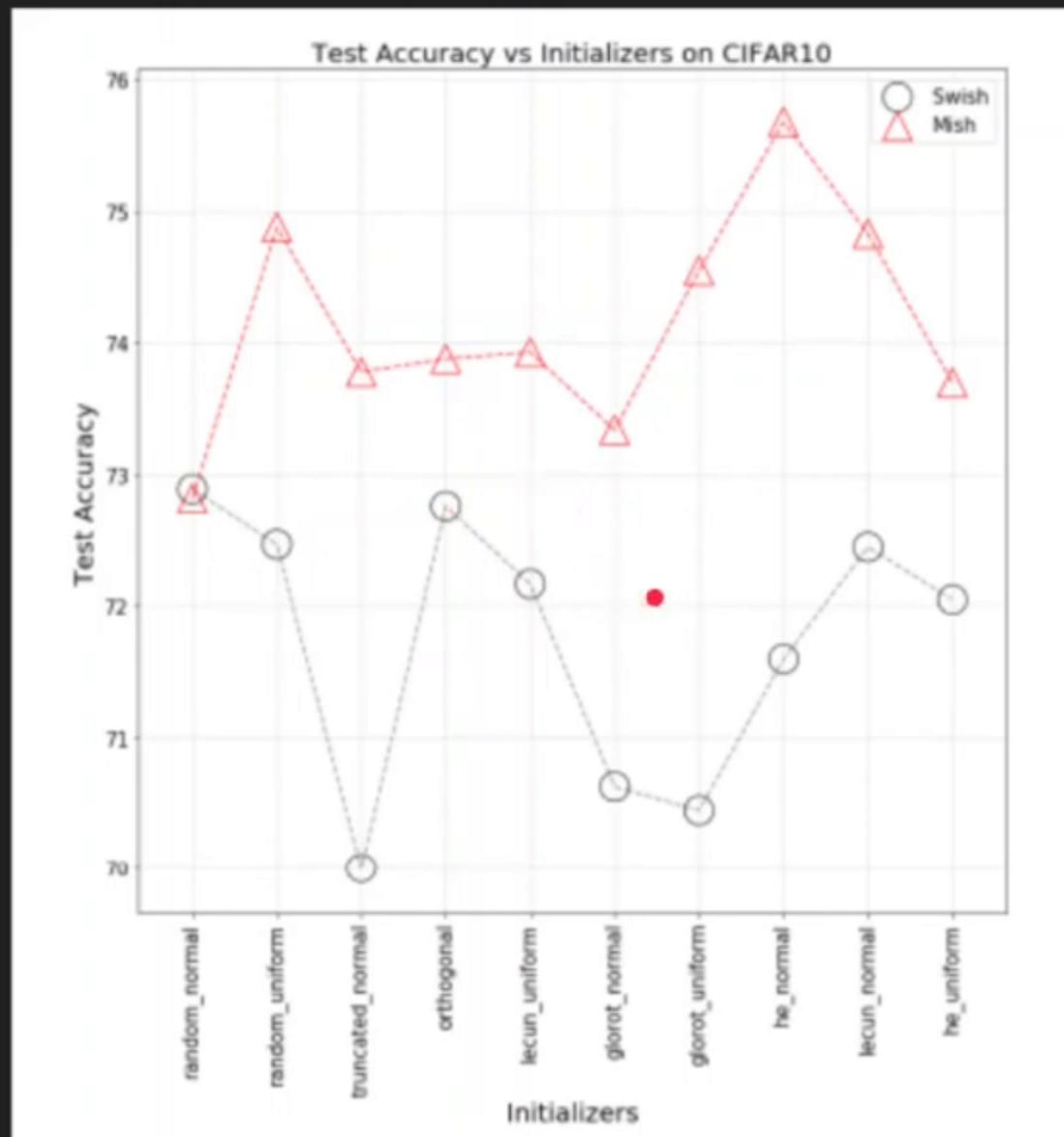
Ablations - Dropout



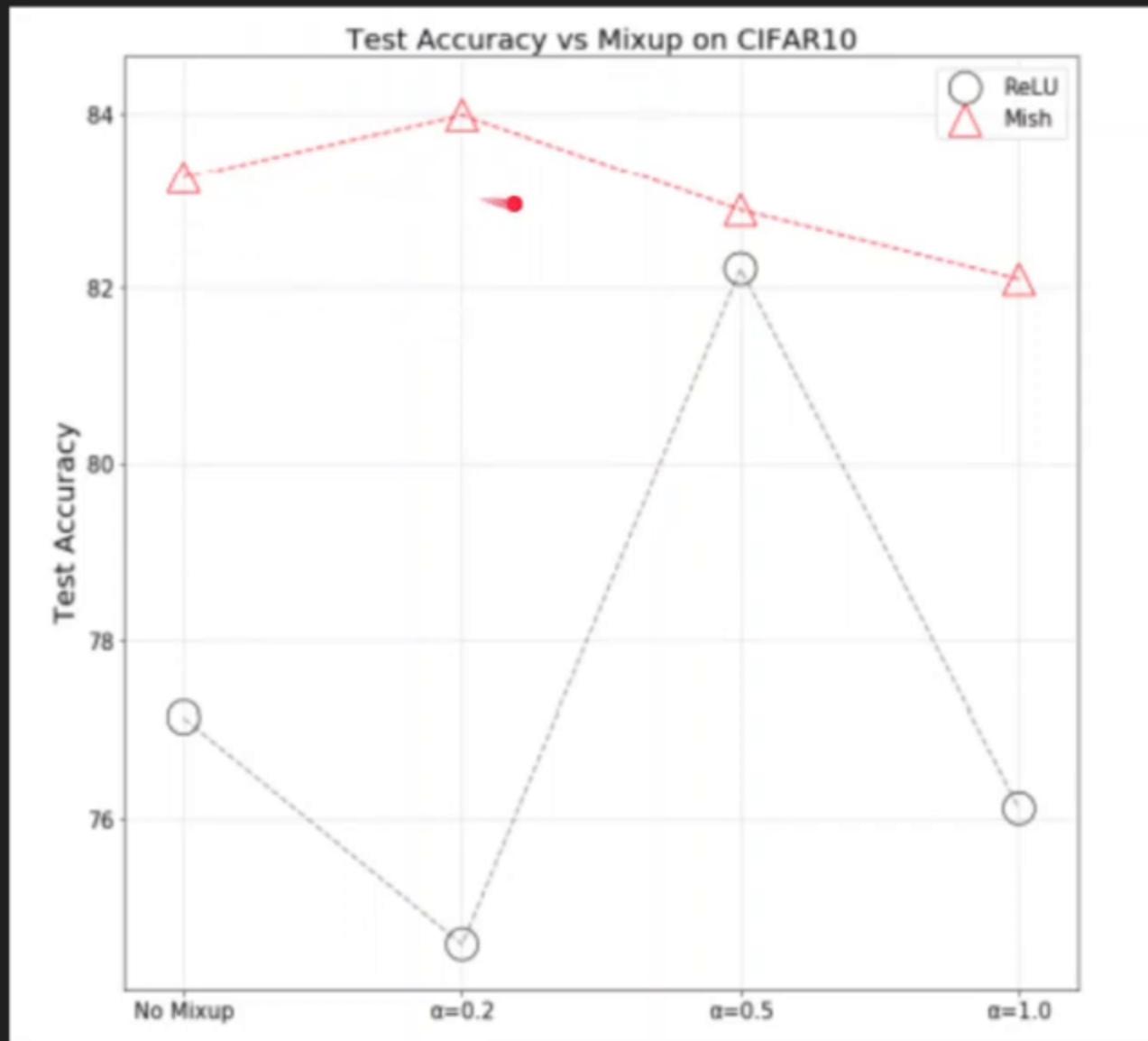
Ablations - Weight Initialization



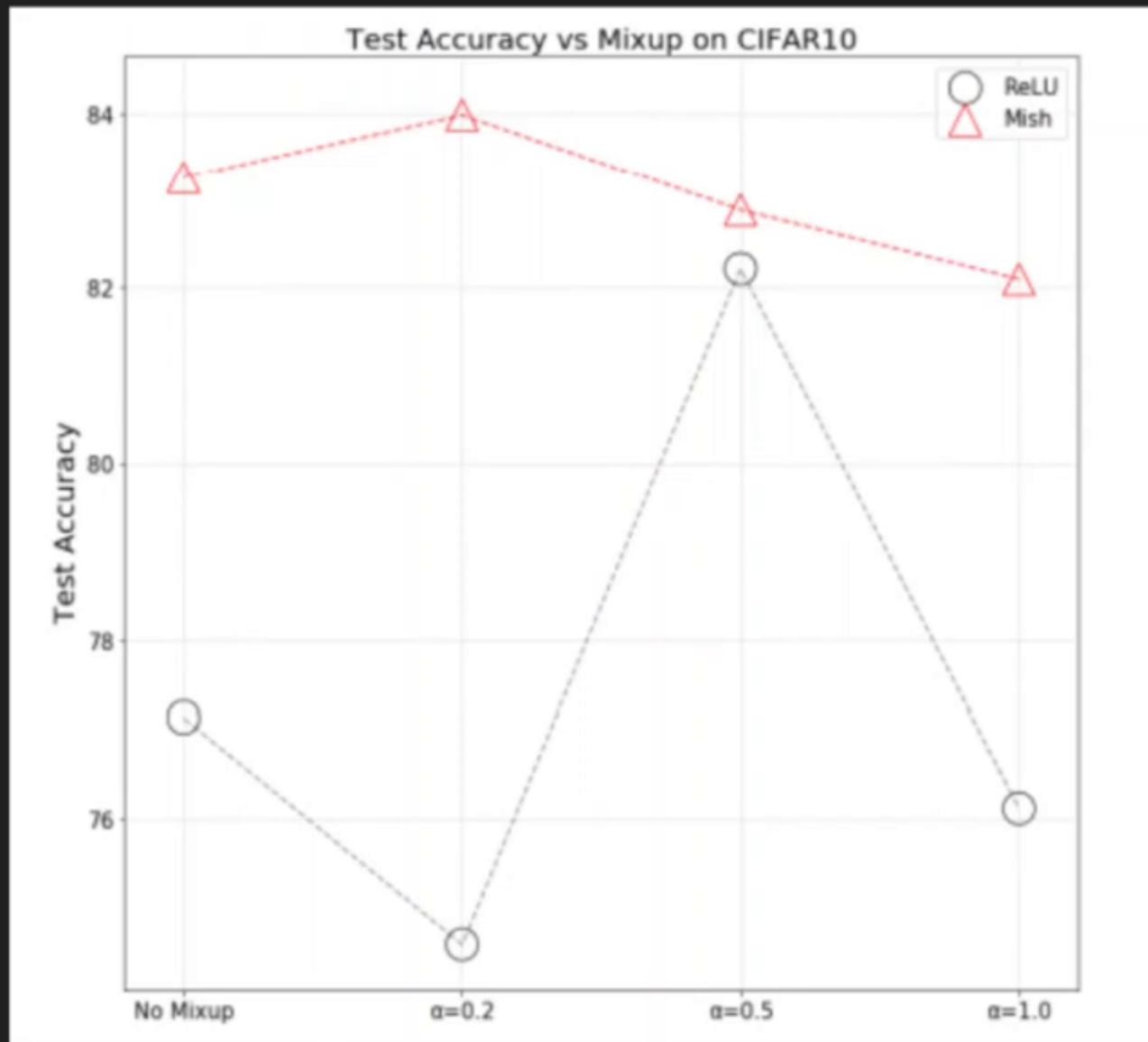
Ablations - Weight Initialization



Ablations - Data Augmentation



Ablations - Data Augmentation



Python Implementation

$$f(x) = x \cdot \tanh(\ln(1 + e^x))$$

```
def mish(x):
    m = x * np.tanh(np.log(1+np.exp(x)))
    return m
```

Mish Activation - Google Slides x Neuron.ipynb - Colaboratory x +

colab.research.google.com/drive/109Wvb6_MPN9nDPOxf4wy2r6Ip7MZz4fs#scrollTo=xi7FuP9hzLp7

Neuron.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings User

+ Code + Text RAM Disk Editing

Swish & Mish Activation

```
import numpy as np
import matplotlib.pyplot as plt

def swish(x):
    s = x * (1/(1+np.exp(-x)))
    return s

def mish(x):
    m = x * np.tanh(np.log(1+np.exp(x)))
    return m
```

```
[ ] z=np.arange(-4,4,0.01)

fig, ax = plt.subplots(figsize=(9, 5))
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.plot(z,swish(z), color="#307EC7", linewidth=3, label="swish")
ax.plot(z,mish(z), color="#9621E2", linewidth=3, label="mish")
ax.legend(loc="upper left", frameon=False)
fig.show()
```

✓ 0s completed at 8:08 PM



Mish Activation - Google Slides x Neuron.ipynb - Colaboratory x +

colab.research.google.com/drive/109Wvb6_MPN9nDPOxf4wy2r6Ip7MZz4fs#scrollTo=TZSjtH_NbYW6

Neuron.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings User

+ Code + Text RAM Disk Editing

```
[2] def swish(x):
    s = x * (1/(1+np.exp(-x)))
    return s

def mish(x):
    m = x * np.tanh(np.log(1+np.exp(x)))
    return m
```

RAM Disk

Editing

z=np.arange(1,4,0.01)

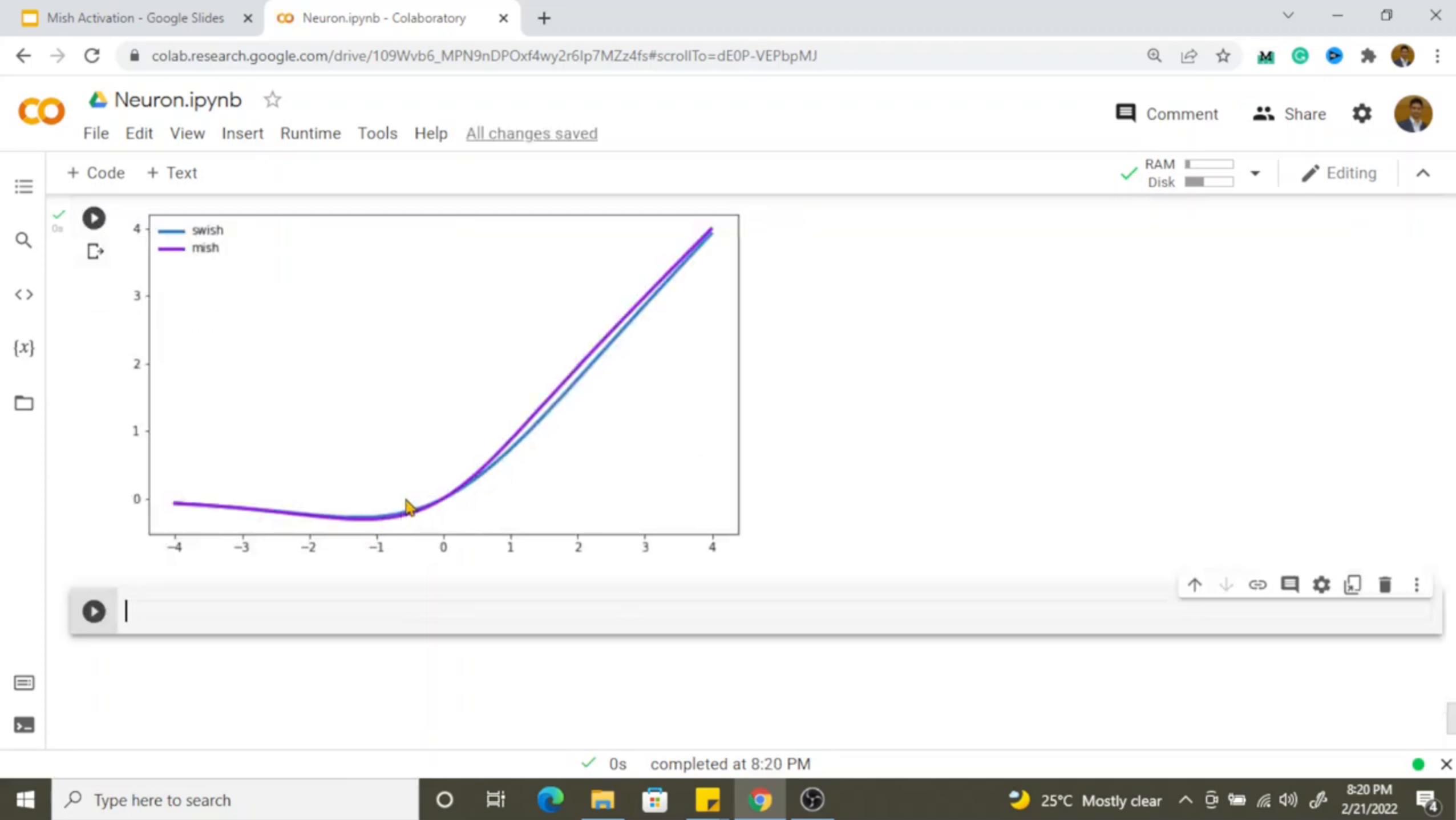
fig, ax = plt.subplots(figsize=(9, 5))
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.plot(z,swish(z), color="#307EC7", linewidth=3, label="swish")
ax.plot(z,mish(z), color="#9621E2", linewidth=3, label="mish")
ax.legend(loc="upper left", frameon=False)
fig.show()

[]

0s completed at 8:19 PM

25°C Mostly clear 8:19 PM 2/21/2022

Type here to search



Mish Activation - Google Slides x Neuron.ipynb - Colaboratory x +

colab.research.google.com/drive/109Wvb6_MPN9nDPOxf4wy2r6Ip7MZz4fs#scrollTo=dE0P-VEPbpMJ

Comment Share Settings User Profile

Neuron.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ RAM Disk ✓ Editing

0s

```
fig, ax = plt.subplots(figsize=(9, 5))
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.plot(z,swish(z), color="#307EC7", linewidth=3, label="swish")
ax.plot(z,mish(z), color="#9621E2", linewidth=3, label="mish")
ax.legend(loc="upper left", frameon=False)
fig.show()
```

{x}

4
3
2
1
0

swish
mish

-4 -3 -2 -1 0 1 2 3 4

✓ 0s completed at 8:20 PM

Type here to search

8:20 PM 25°C Mostly clear 2/21/2022