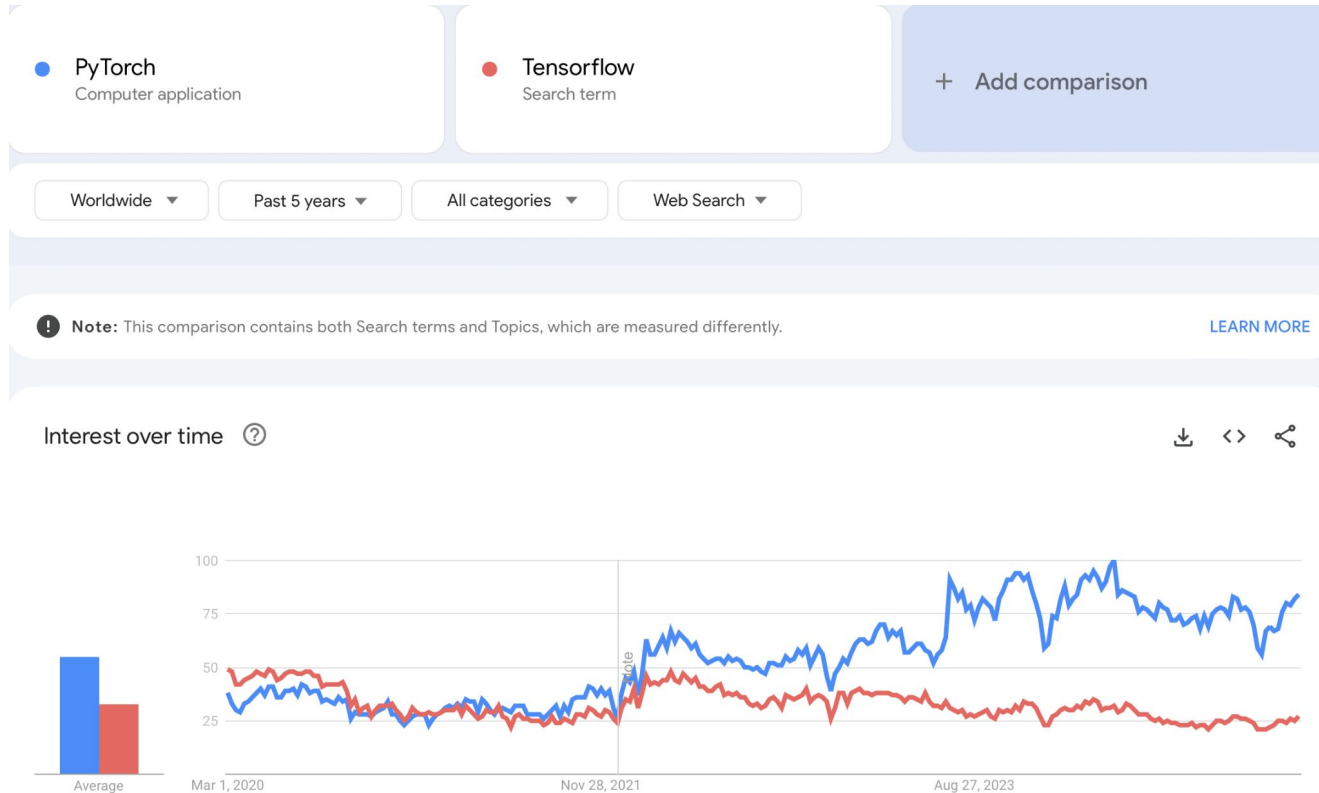


Convolutional Neural Networks

Marta Szuwarska, Mateusz Nizwantowski



Framework choice



Baseline Model

Layer (type:depth-idx)	Output Shape	Param #
BaselineModel	[64, 10]	--
└Sequential: 1-1	[64, 32, 16, 16]	--
└Conv2d: 2-1	[64, 32, 32, 32]	896
└ReLU: 2-2	[64, 32, 32, 32]	--
└Conv2d: 2-3	[64, 32, 32, 32]	9,248
└ReLU: 2-4	[64, 32, 32, 32]	--
└MaxPool2d: 2-5	[64, 32, 16, 16]	--
└Sequential: 1-2	[64, 32, 8, 8]	--
└Conv2d: 2-6	[64, 32, 16, 16]	9,248
└ReLU: 2-7	[64, 32, 16, 16]	--
└Conv2d: 2-8	[64, 32, 16, 16]	9,248
└ReLU: 2-9	[64, 32, 16, 16]	--
└MaxPool2d: 2-10	[64, 32, 8, 8]	--
└Sequential: 1-3	[64, 10]	--
└Flatten: 2-11	[64, 2048]	--
└Dropout: 2-12	[64, 2048]	--
└Linear: 2-13	[64, 10]	20,490
Total params: 49,130		
Trainable params: 49,130		
Non-trainable params: 0		
Total mult-adds (Units.MEGABYTES): 969.15		
Input size (MB): 0.79		
Forward/backward pass size (MB): 41.95		
Params size (MB): 0.20		
Estimated Total Size (MB): 42.93		

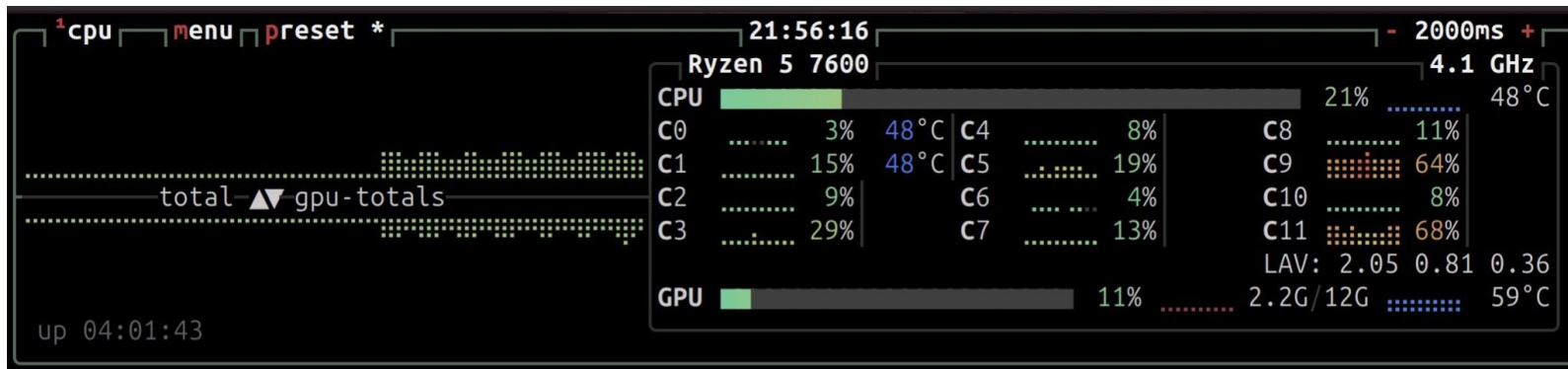
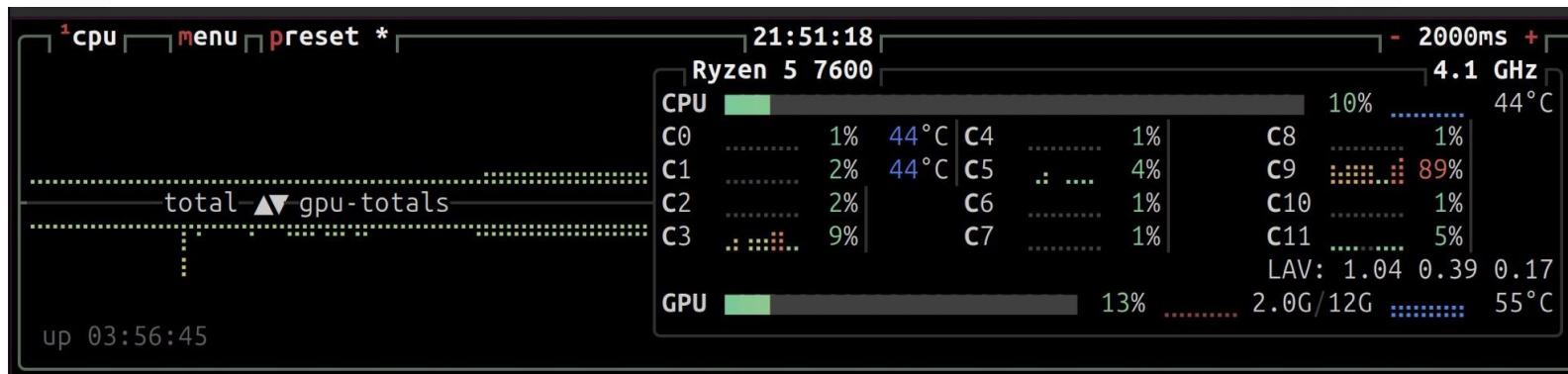


Enhanced Model

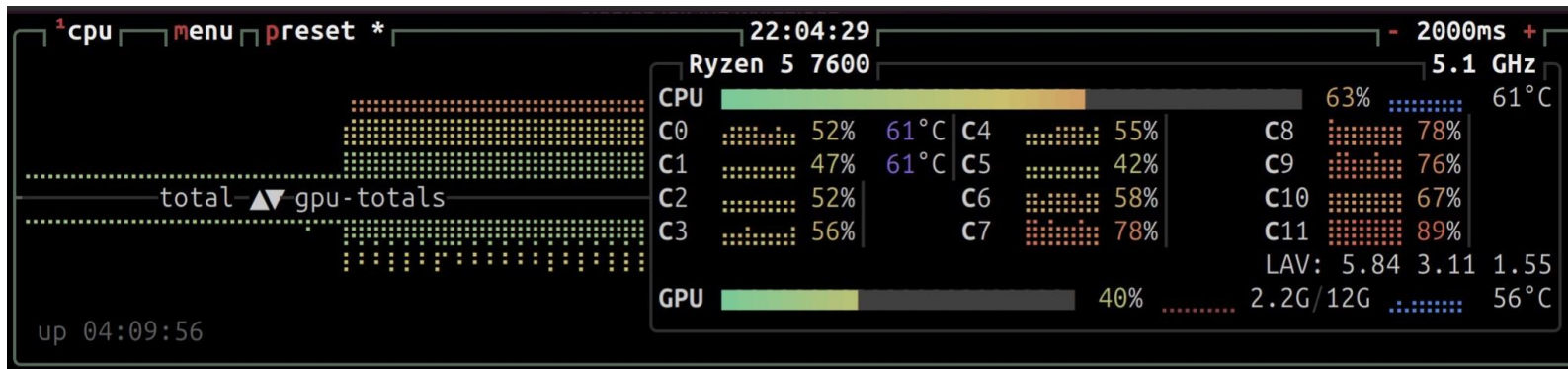
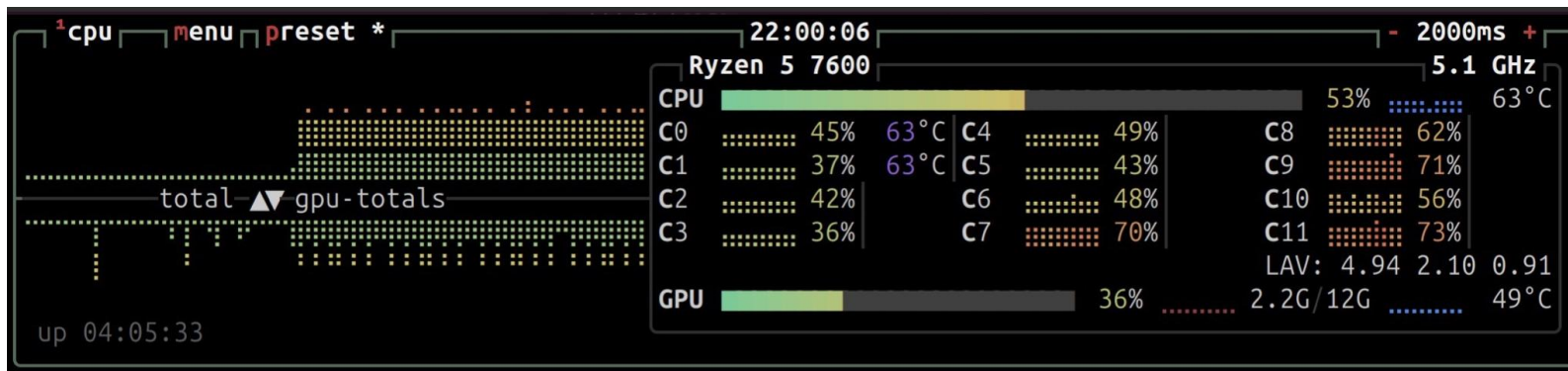
Layer (type:depth-idx)	Output Shape	Param #
EnhancedModel	[64, 10]	--
└Sequential: 1-1	[64, 64, 16, 16]	--
└Conv2d: 2-1	[64, 64, 32, 32]	1,792
└ReLU: 2-2	[64, 64, 32, 32]	--
└BatchNorm2d: 2-3	[64, 64, 32, 32]	128
└Conv2d: 2-4	[64, 64, 32, 32]	36,928
└ReLU: 2-5	[64, 64, 32, 32]	--
└BatchNorm2d: 2-6	[64, 64, 32, 32]	128
└MaxPool2d: 2-7	[64, 64, 16, 16]	--
└Sequential: 1-2	[64, 128, 8, 8]	--
└Conv2d: 2-8	[64, 128, 16, 16]	73,856
└ReLU: 2-9	[64, 128, 16, 16]	--
└BatchNorm2d: 2-10	[64, 128, 16, 16]	256
└Conv2d: 2-11	[64, 128, 16, 16]	147,584
└ReLU: 2-12	[64, 128, 16, 16]	--
└BatchNorm2d: 2-13	[64, 128, 16, 16]	256
└MaxPool2d: 2-14	[64, 128, 8, 8]	--
└Sequential: 1-3	[64, 256, 4, 4]	--
└Conv2d: 2-15	[64, 256, 8, 8]	295,168
└ReLU: 2-16	[64, 256, 8, 8]	--
└BatchNorm2d: 2-17	[64, 256, 8, 8]	512
└Conv2d: 2-18	[64, 256, 8, 8]	590,080
└ReLU: 2-19	[64, 256, 8, 8]	--
└BatchNorm2d: 2-20	[64, 256, 8, 8]	512
└MaxPool2d: 2-21	[64, 256, 4, 4]	--
└AdaptiveAvgPool2d: 1-4	[64, 256, 1, 1]	--
└Sequential: 1-5	[64, 10]	--
└Flatten: 2-22	[64, 256]	--
└Dropout: 2-23	[64, 256]	--
└Linear: 2-24	[64, 10]	2,570
=====		
Total params: 1,149,770		
Trainable params: 1,149,770		
Non-trainable params: 0		
Total mult-adds (Units.GIGABYTES): 9.79		
=====		
Input size (MB): 0.79		
Forward/backward pass size (MB): 234.89		
Params size (MB): 4.60		
Estimated Total Size (MB): 240.27		
=====		



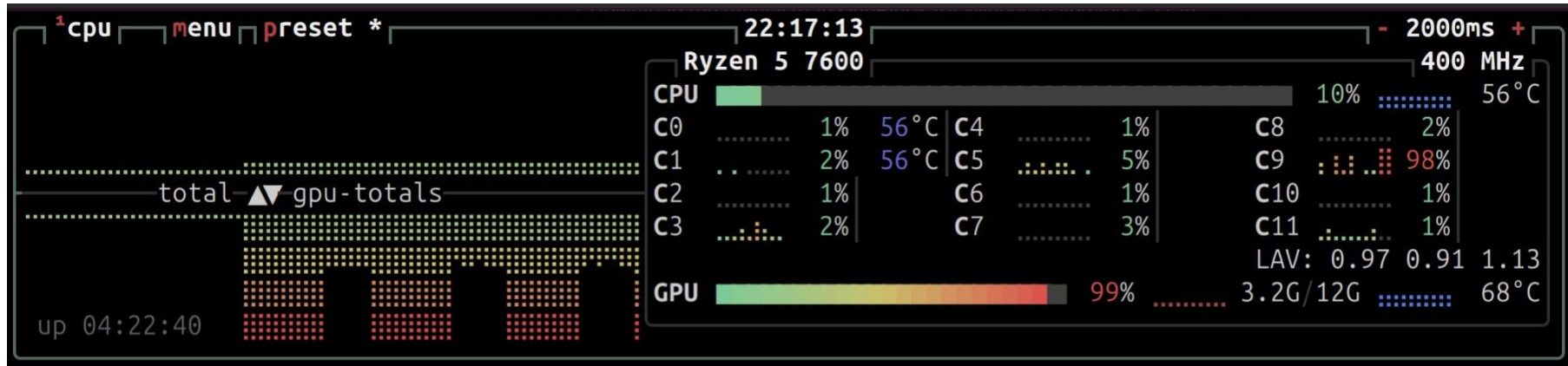
Tweaking Data Loader



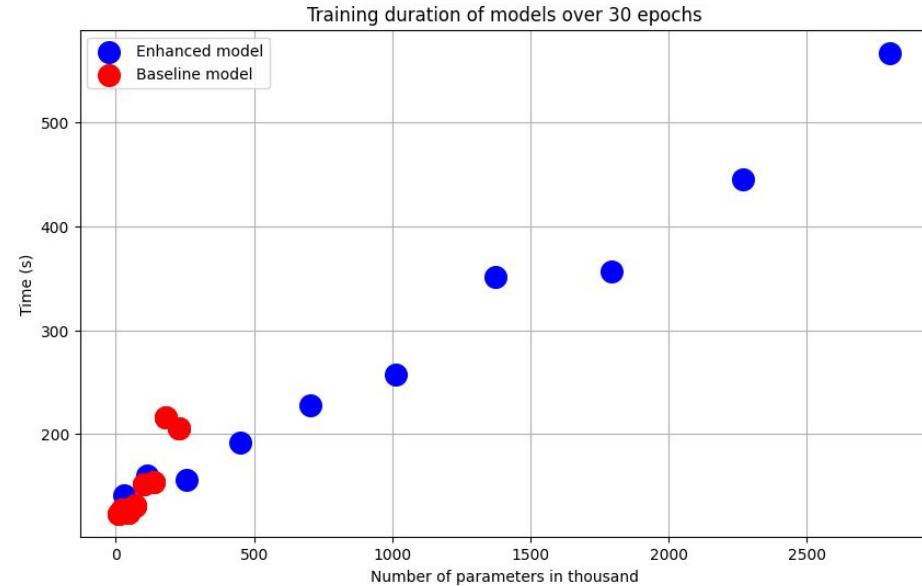
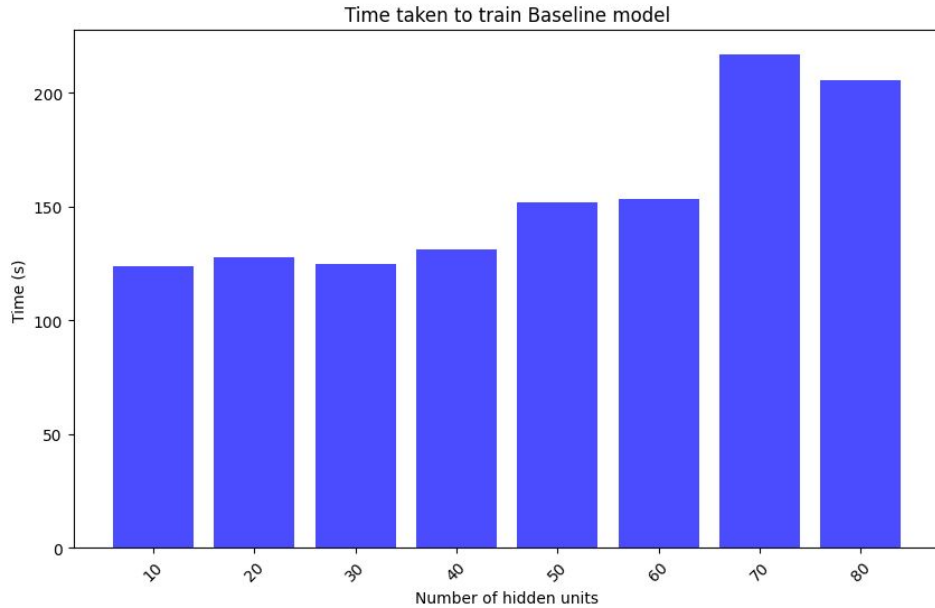
Tweaking Data Loader



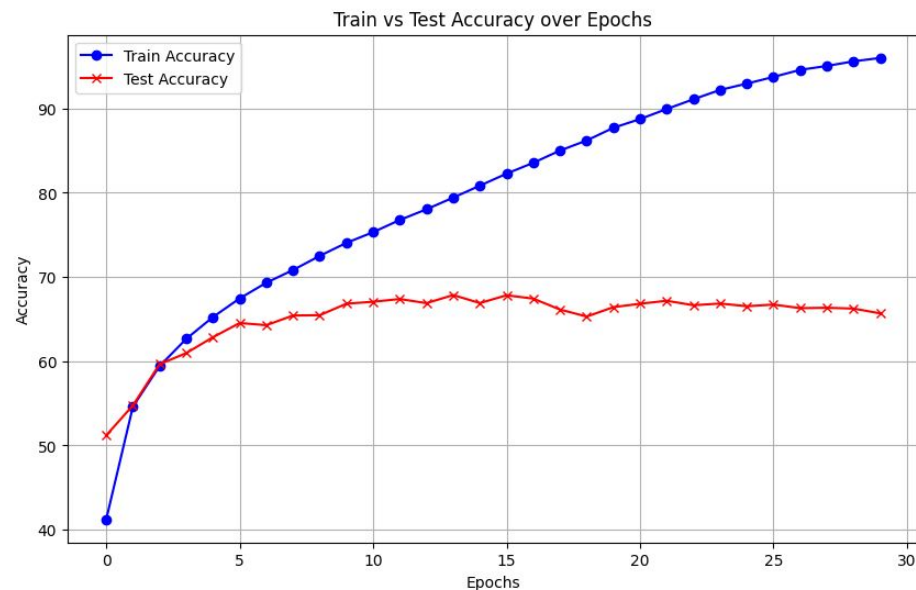
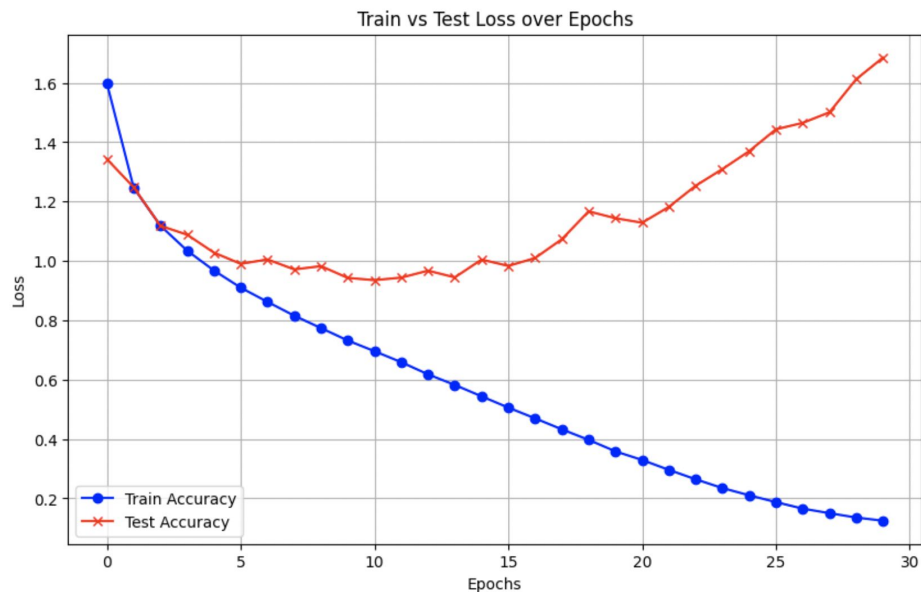
Tweaking Data Loader



Time measurements

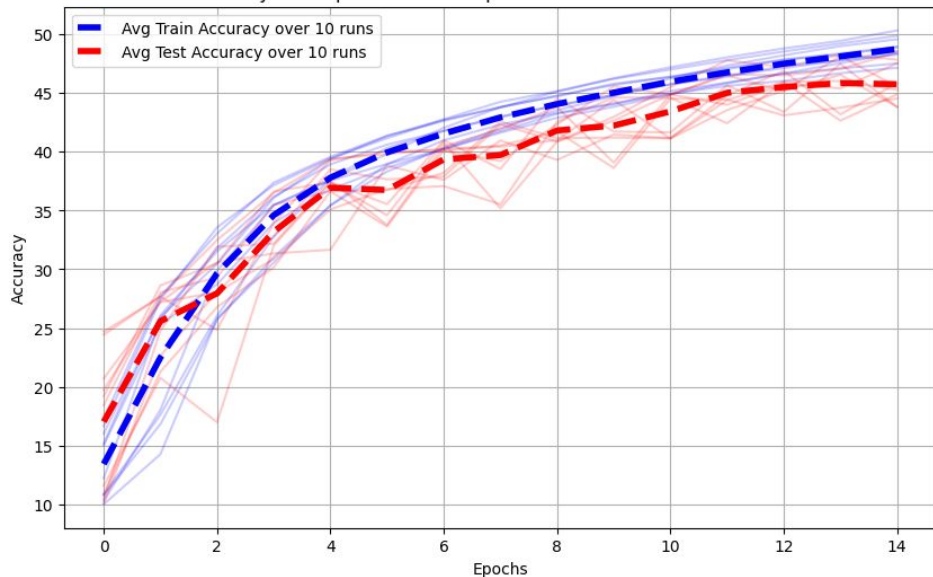


Plots - visualization of overfitting

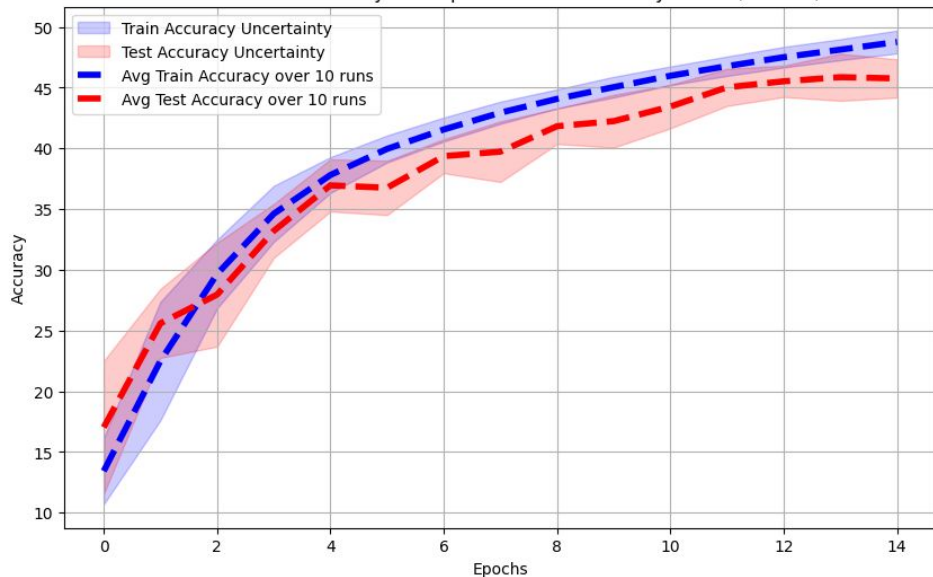


Plots - how reliable are measurements

Train vs Test Accuracy over Epochs and multiple runs on baseline model with 10 hidden units

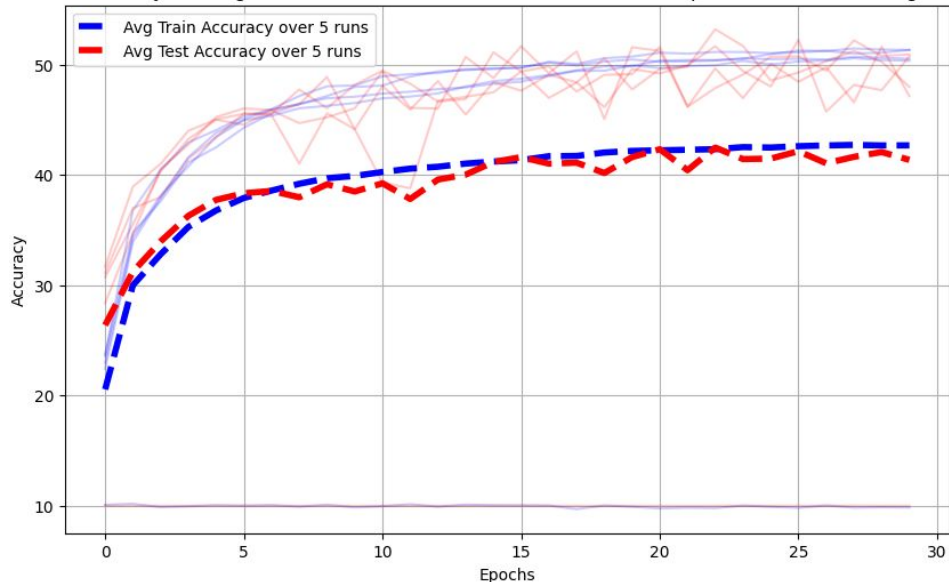


Train vs Test Accuracy over Epochs with Uncertainty Bands (10 runs)

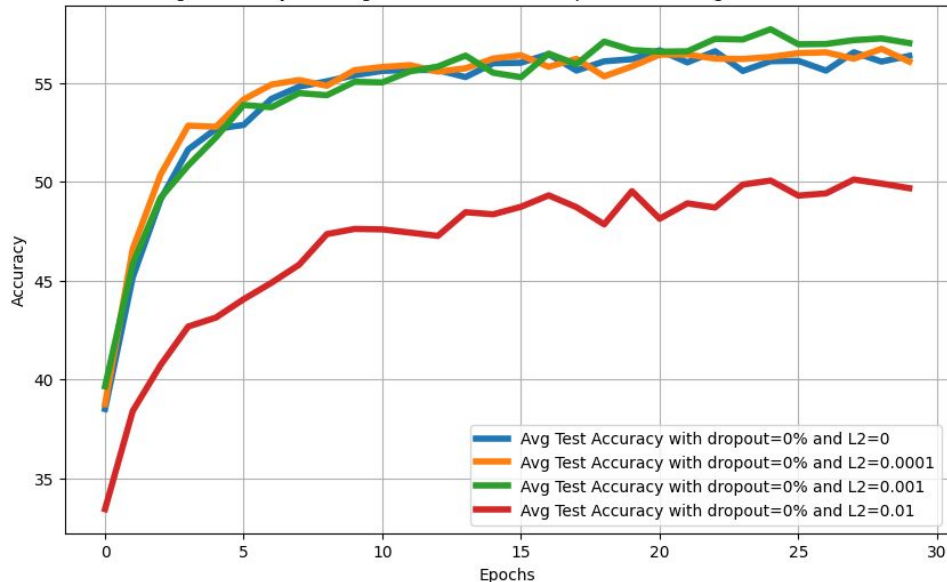


Regularization

Accuracy convergence of individual runs for model with 75% dropout and 0.01 L2 strength

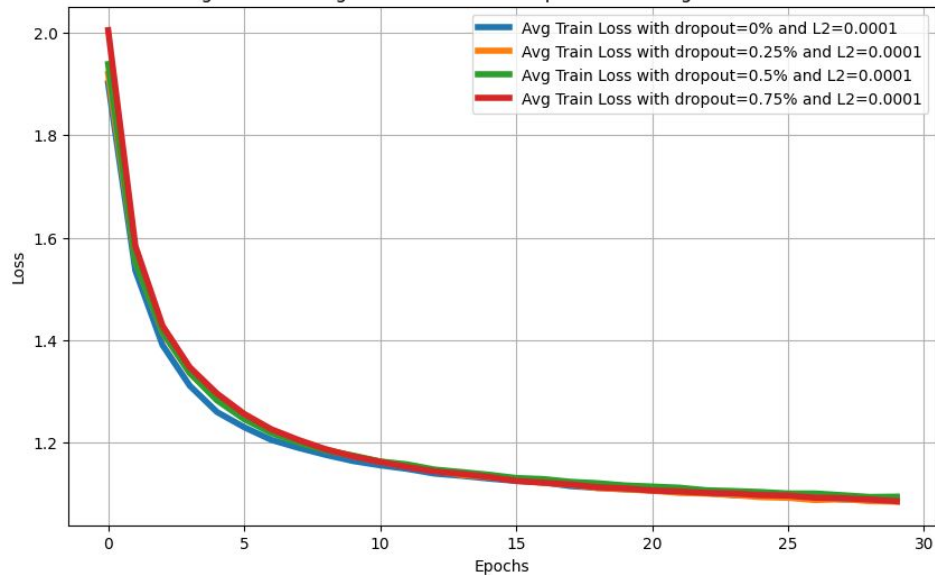


Average accuracy convergence for different dropout and L2 regularization values

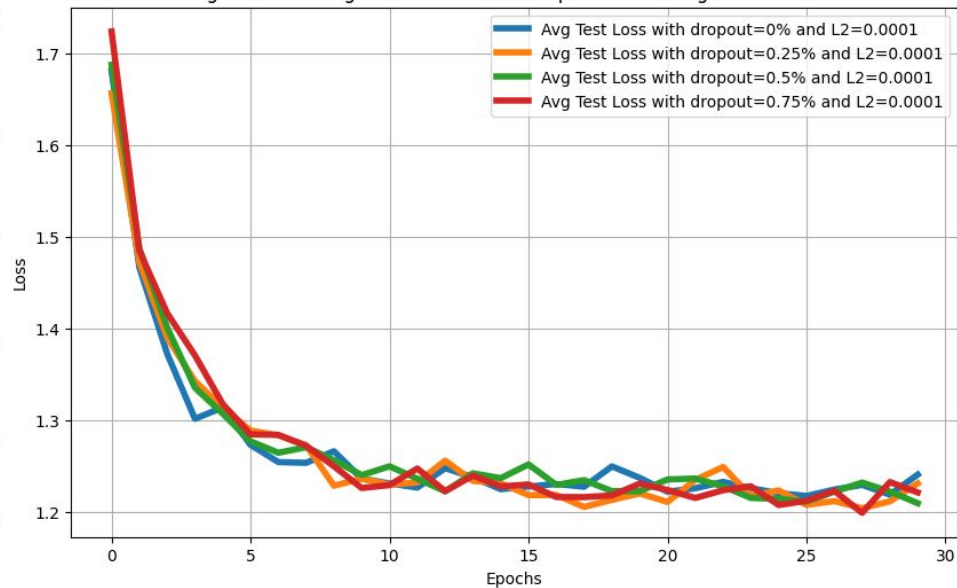


Regularization

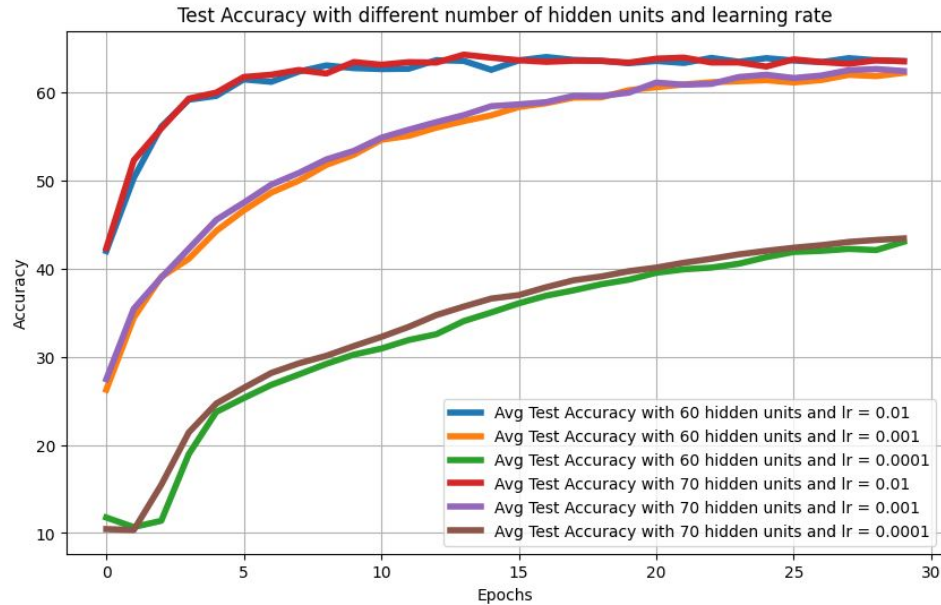
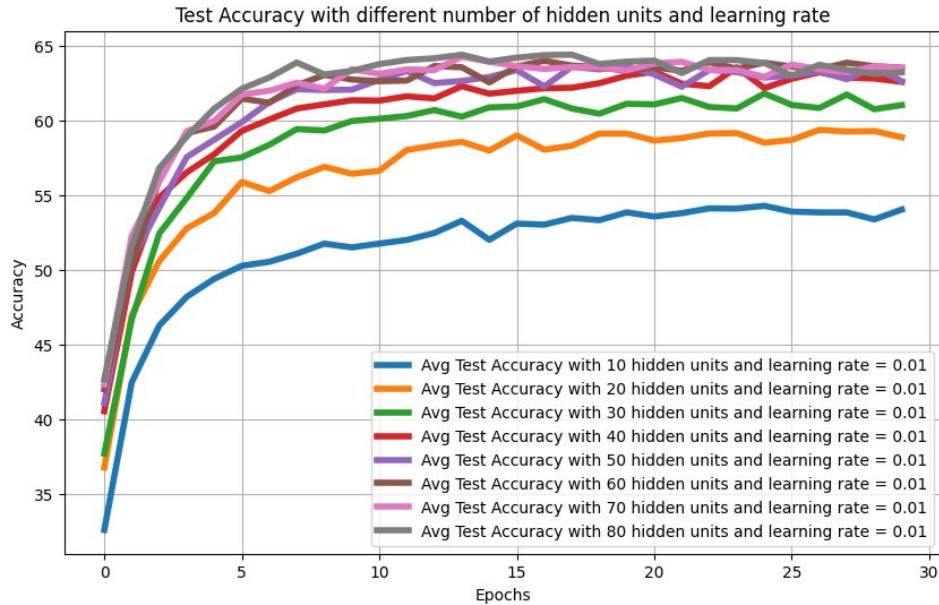
Average loss convergence for different dropout and L2 regularization values



Average loss convergence for different dropout and L2 regularization values

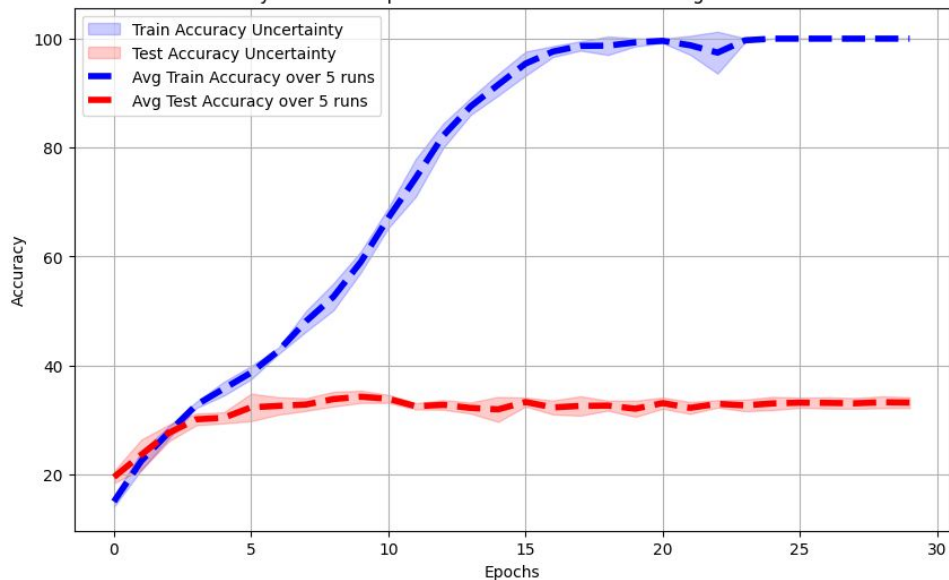


Learning rate and number of hidden units

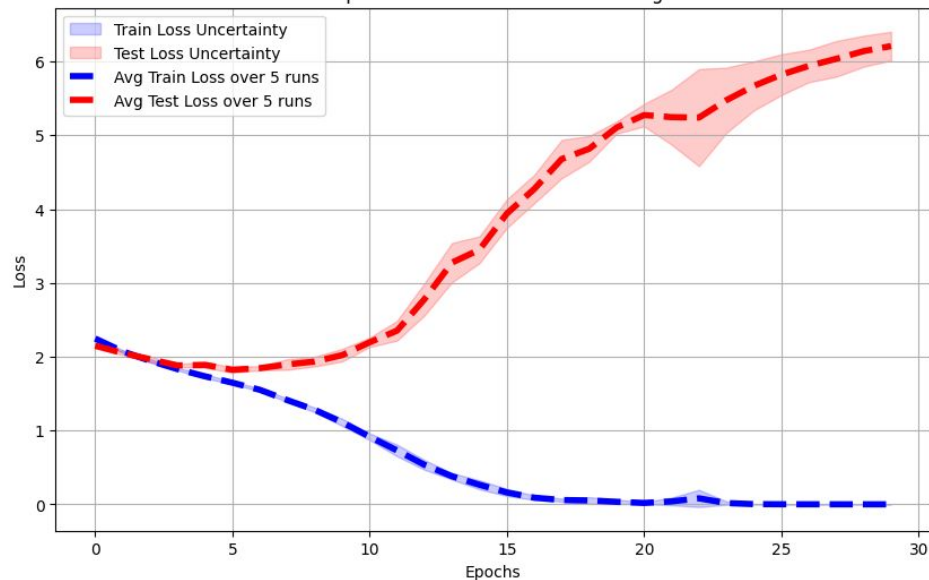


Few-shot learning - basic implementation

Accuracy for basic implementation of few shot learning with $\text{lr}=0.001$

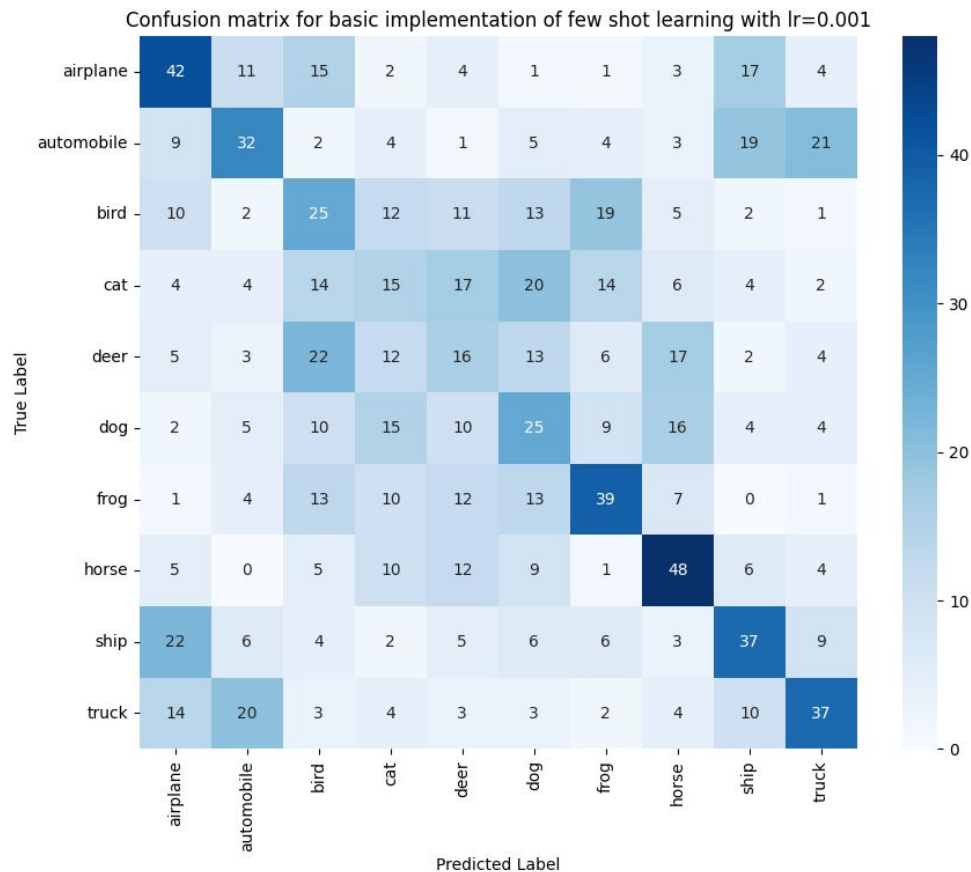


Loss for basic implementation of few shot learning with $\text{lr}=0.001$



Few-shot learning

- basic implementation



Few-shot learning - pretrained ResNet-18

No.	learning rate	optimizer	unfreezed layers	cosine classifier	train acc	test acc
1	0.01	Adam	2	no	94%	41%
2	0.001	Adam	2	no	100%	44%
3	0.0001	Adam	2	no	100%	41%
4	0.001	SGD	2	no	100%	43%
5	0.001	Adam	3	no	93%	43%
6	0.001	Adam	1	no	94%	36%
7	0.001	Adam	2	yes	90%	44%

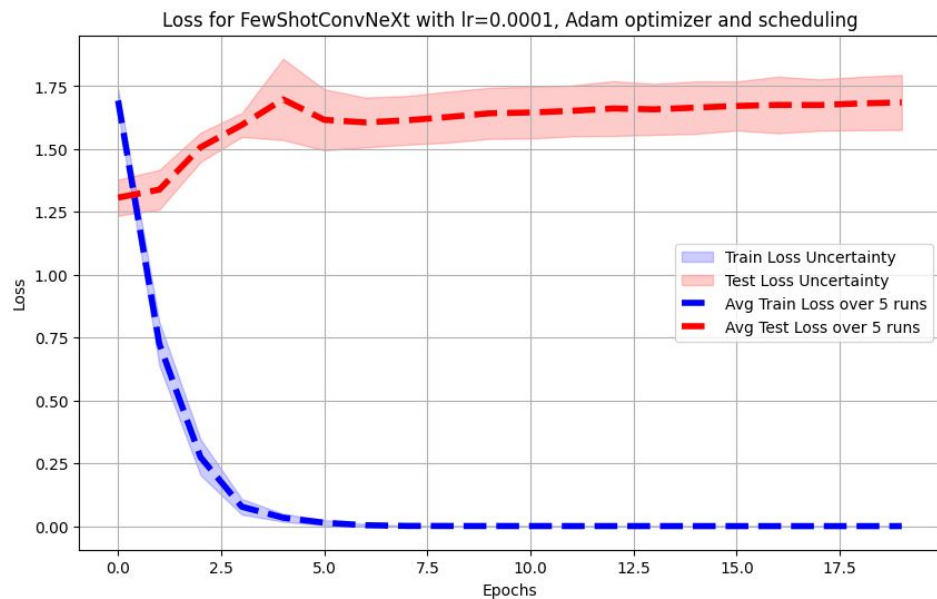
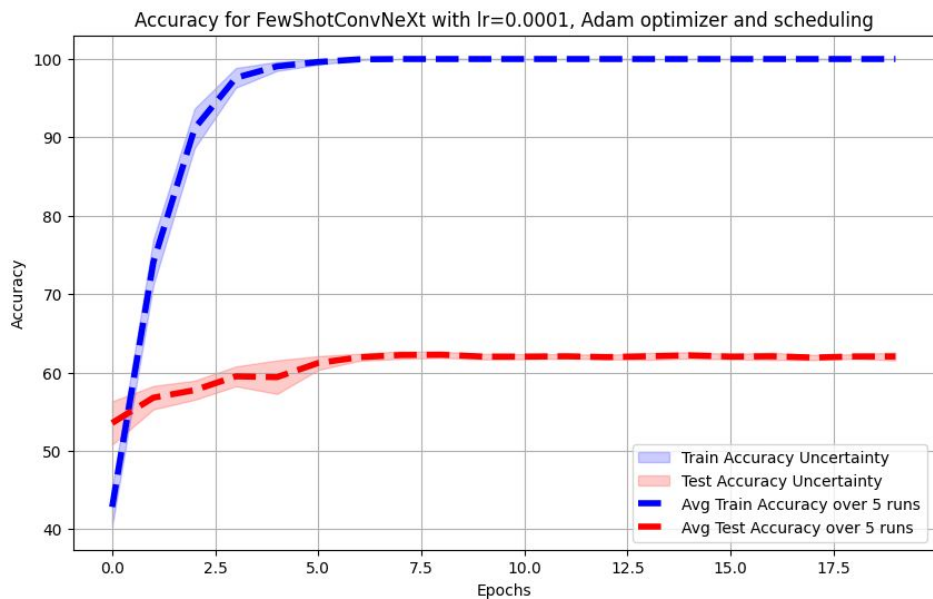


Few-shot learning - pretrained ConvNeXt-Tiny

No.	initial learning rate	optimizer	scheduling	train acc	test acc
1	0.001	Adam	no	98%	50%
2	0.0001	Adam	no	100%	60%
3	0.00001	Adam	no	100%	59%
4	0.0001	Adam	yes	100%	63%
5	0.0001	SGD	no	100%	60%
6	0.0001	SGD	yes	98%	59%



Few-shot learning - pretrained ConvNeXt-Tiny



Few-shot learning - other models

No.	method	learning rate	optimizer	scheduling	train acc	test acc
1	prototypical	0.0001	Adam	yes	47%	46%
2	siamese network	0.0001	Adam	no	49%	51%
3	maml	0.001 + 0.0001	Adam	no	100%	14%

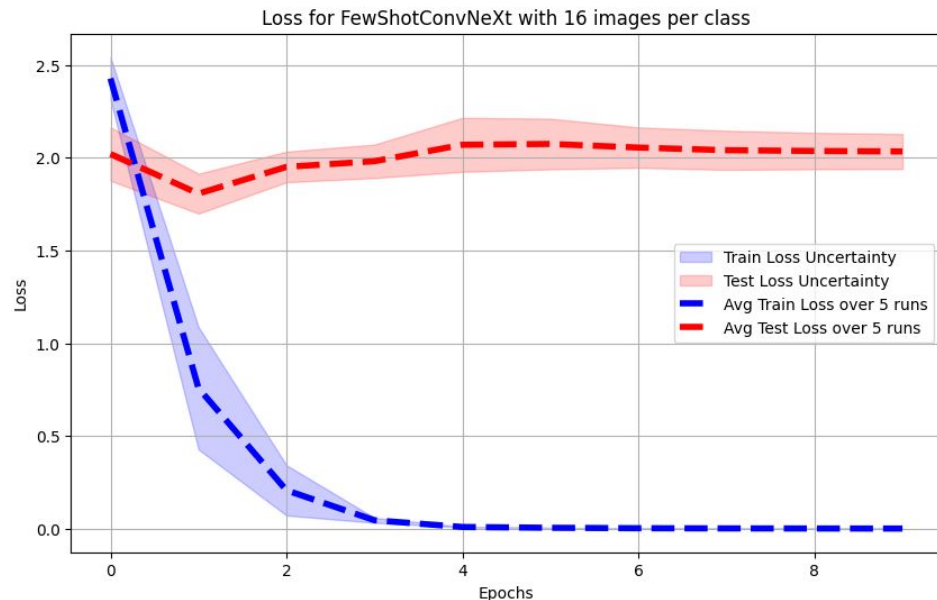
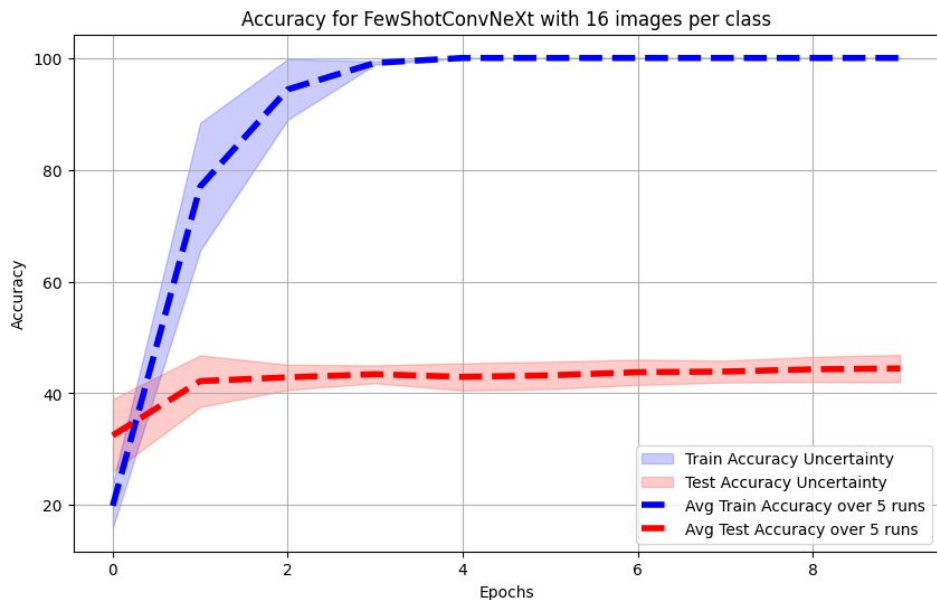


Few-shot learning - regularization

No.	dropout	weight decay	train acc	test acc
1	-	-	100%	61%
2	0.25	-	100%	60%
3	0.5	-	100%	61%
4	0.75	-	100%	59%
5	-	1e-5	100%	60%
6	-	1e-4	100%	61%
7	-	1e-3	100%	60%
8	-	1e-2	100%	59%
9	-	1e-1	92%	54%
10	-	1e-0	51%	47%



Few-shot learning - 16 images per class



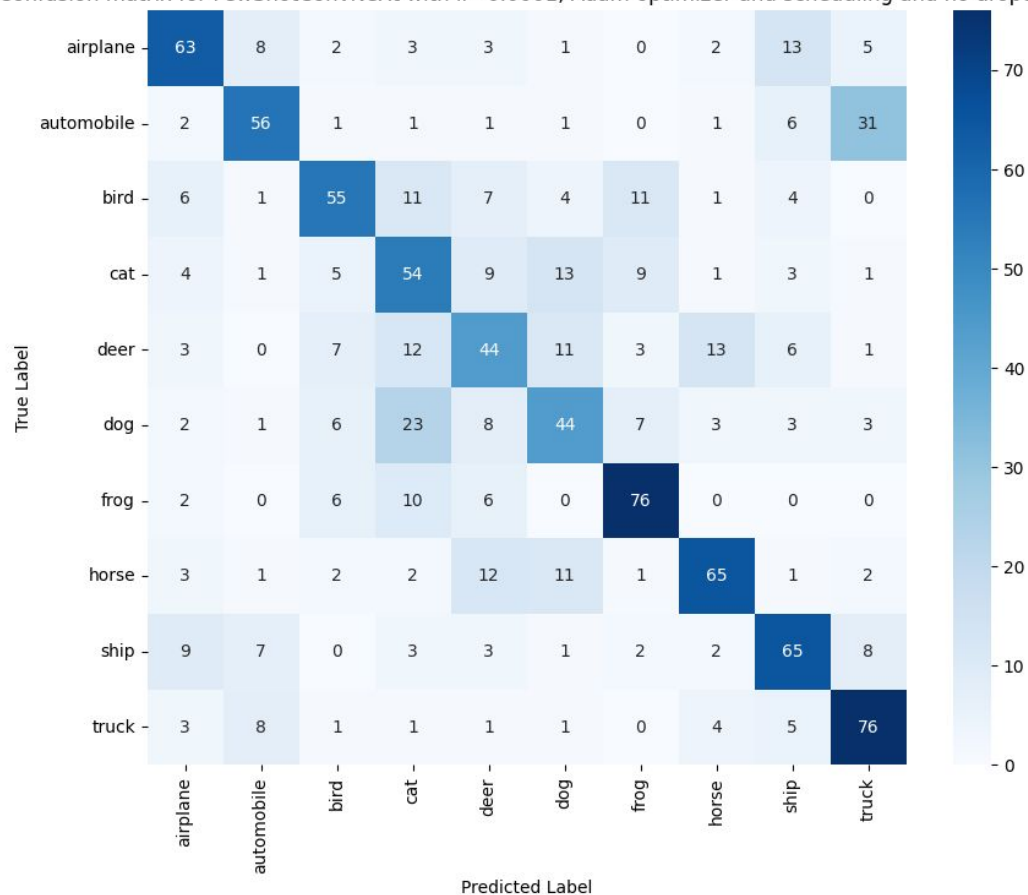
Few-shot learning - augmentation

No.	augmentation	train acc	test acc
1	flip	100%	61%
2	rotation	100%	55%
3	jitter	100%	59%
4	mixed	100%	54%
5	cutmix	100%	39%



Few-shot learning - pretrained ConvNeXt-Tiny

Confusion matrix for FewShotConvNeXt with $\text{lr}=0.0001$, Adam optimizer and scheduling and no dropout



Thank you for your attention!

Are there any questions?



Sources

1. [CNN Explainer](#)
2. [Papers With Code](#)
3. [Few-shot learning \(1\)](#)
4. [Few-shot learning \(2\)](#)

