

# MNIST Training

Document Author

## Contents

<b>1</b>	<b>Summary</b>	<b>1</b>
<b>2</b>	<b>Training reports</b>	<b>2</b>
2.1	Model 1: ConvNet2layers . . . . .	2
2.2	Model 2: ConvNet2layers . . . . .	4
2.3	Model 3: MLP2layers . . . . .	6
2.4	Model 4: MLP2layers . . . . .	8
2.5	Model 5: MLP5layers . . . . .	10
<b>3</b>	<b>Model Architectures</b>	<b>12</b>
3.1	MLP2layers . . . . .	12
3.2	MLP5layers . . . . .	13
3.3	ConvNet2layers . . . . .	14

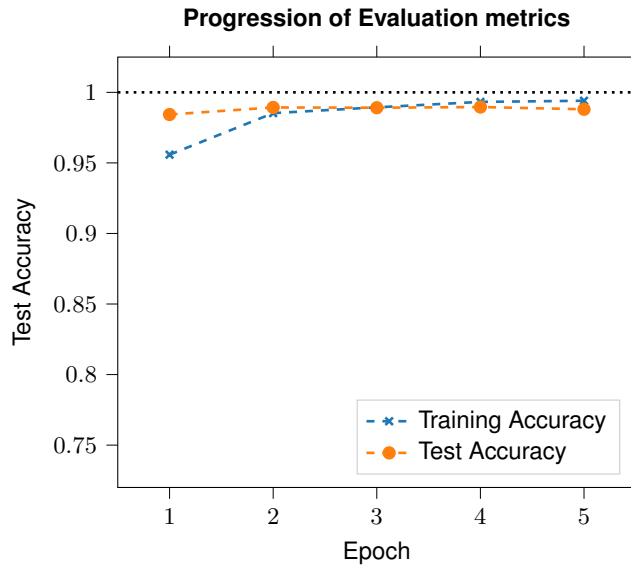
## 1 Summary

Nº	Model name	#Parameters	#Epochs	Batch size	Test Acc.	Training Acc.
1	ConvNet2layers	1 199 882	5	64	98.8 %	99.41 %
2	ConvNet2layers	1 199 882	5	64	98.79 %	99.4 %
3	MLP2layers	669 706	5	64	91.56 %	90.83 %
4	MLP2layers	669 706	5	64	90.62 %	90.83 %
5	MLP5layers	1 457 674	5	64	92.23 %	91.58 %

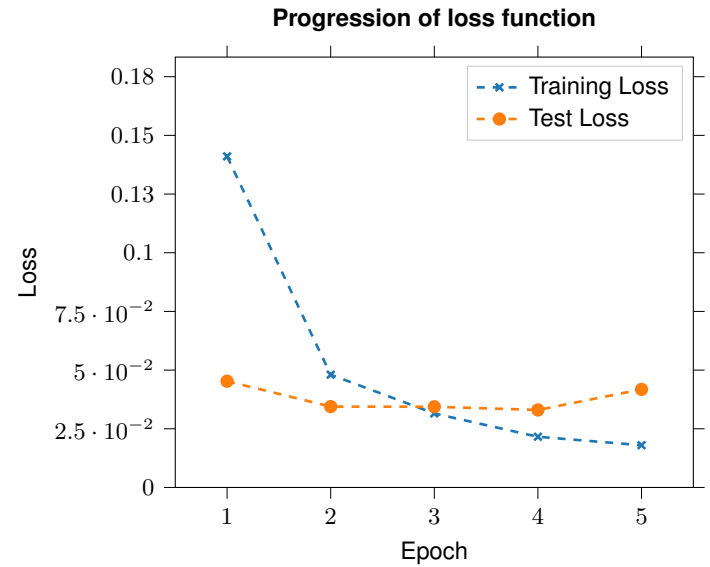
## 2 Training reports

### 2.1 Model 1: ConvNet2layers

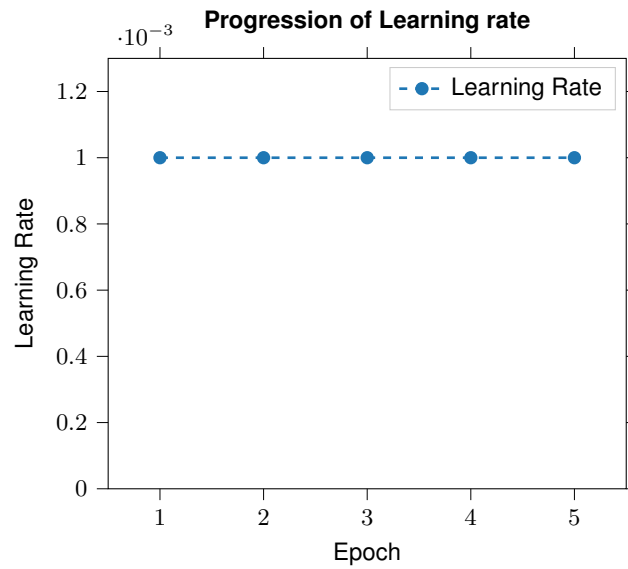
Training history See Figure 1.



(a) Accuracy learning process for model 1.



(b) Loss learning process for model 1.



(c) Learning rate per epoch for model 1.

Figure 1: Training and evaluation metrics for model 1.

## Dataset

**Name** MNIST

**Train-Test-Dev split:** *Training set:* 60000, *Test set:* 10000, *Dev set:* 0,

**Image size** [28, 28]

## Training

**Number of epochs** 5

**Optimizer** Adam (Kingma et al., 2015)

**Learning Rate** 0.0010000000474974513

**Beta 1** 0.89999999761581421

**Beta 2** 0.9990000128746033

**Decay** 0.0

**Epsilon** 1e-07

**Amsgrad** False

**Loss** Categorical crossentropy

**Batch size** 64

**Shuffle** Yes

**Training time** 54 sec

## Platform

**Weights exported to path** weights\ConvNet2layers\_5ep\_MNIST.h5

**Device used** GPU (GeForce GTX 1060 6GB)

**CPU** Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, X86\_64

**Python Version** 3.7.2.final.0 (64 bit)

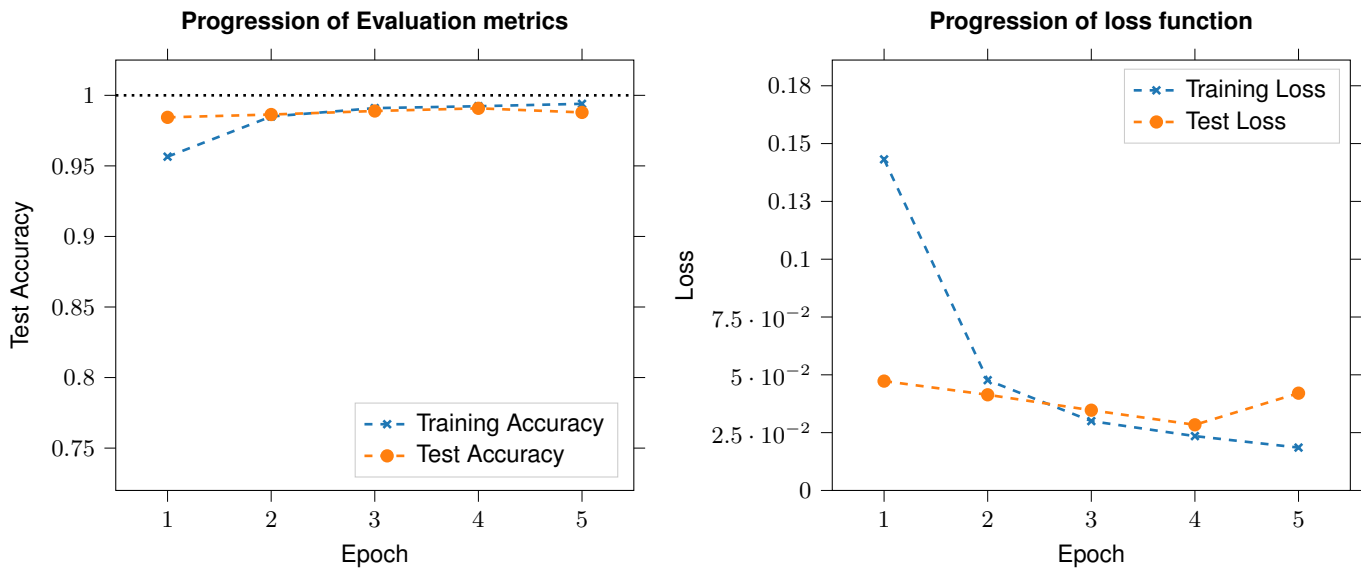
**Keras Version** 2.2.5 (Backend: tensorflow)

**Tensorflow Version** 1.14.0

**Timestamp** 26.09.2019 at 11:15

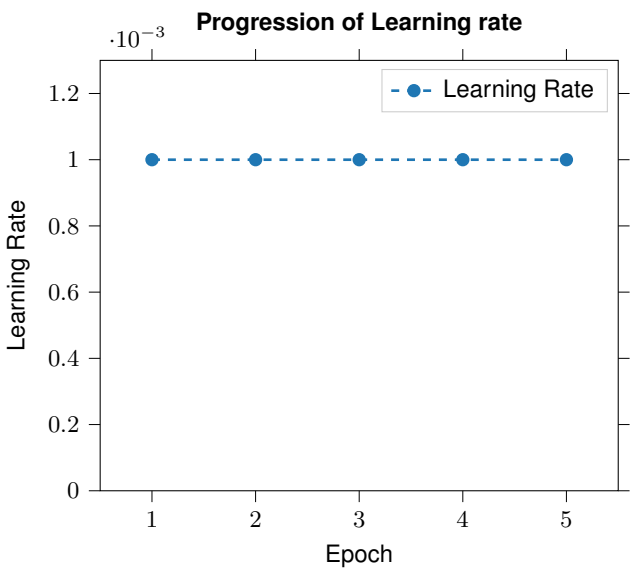
## 2.2 Model 2: ConvNet2layers

Training history See Figure 2.



(a) Accuracy learning process for model 2.

(b) Loss learning process for model 2.



(c) Learning rate per epoch for model 2.

Figure 2: Training and evaluation metrics for model 2.

### Dataset

Name MNIST

**Train-Test-Dev split:** *Training set:* 60000, *Test set:* 10000, *Dev set:* 0,

**Image size** [28, 28]

## Training

**Number of epochs** 5

**Optimizer** Adam (Kingma et al., 2015)

**Learning Rate** 0.0010000000474974513

**Beta 1** 0.89999999761581421

**Beta 2** 0.9990000128746033

**Decay** 0.0

**Epsilon** 1e-07

**Amsgrad** False

**Loss** Categorical crossentropy

**Batch size** 64

**Shuffle** Yes

**Training time** 54 sec

## Platform

**Weights exported to path** weights\ConvNet2layers\_5ep\_MNIST.h5

**Device used** GPU (GeForce GTX 1060 6GB)

**CPU** Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, X86\_64

**Python Version** 3.7.2.final.0 (64 bit)

**Keras Version** 2.2.5 (Backend: tensorflow)

**Tensorflow Version** 1.14.0

**Timestamp** 26.09.2019 at 11:17

### 2.3 Model 3: MLP2layers

Training history See Figure 3.

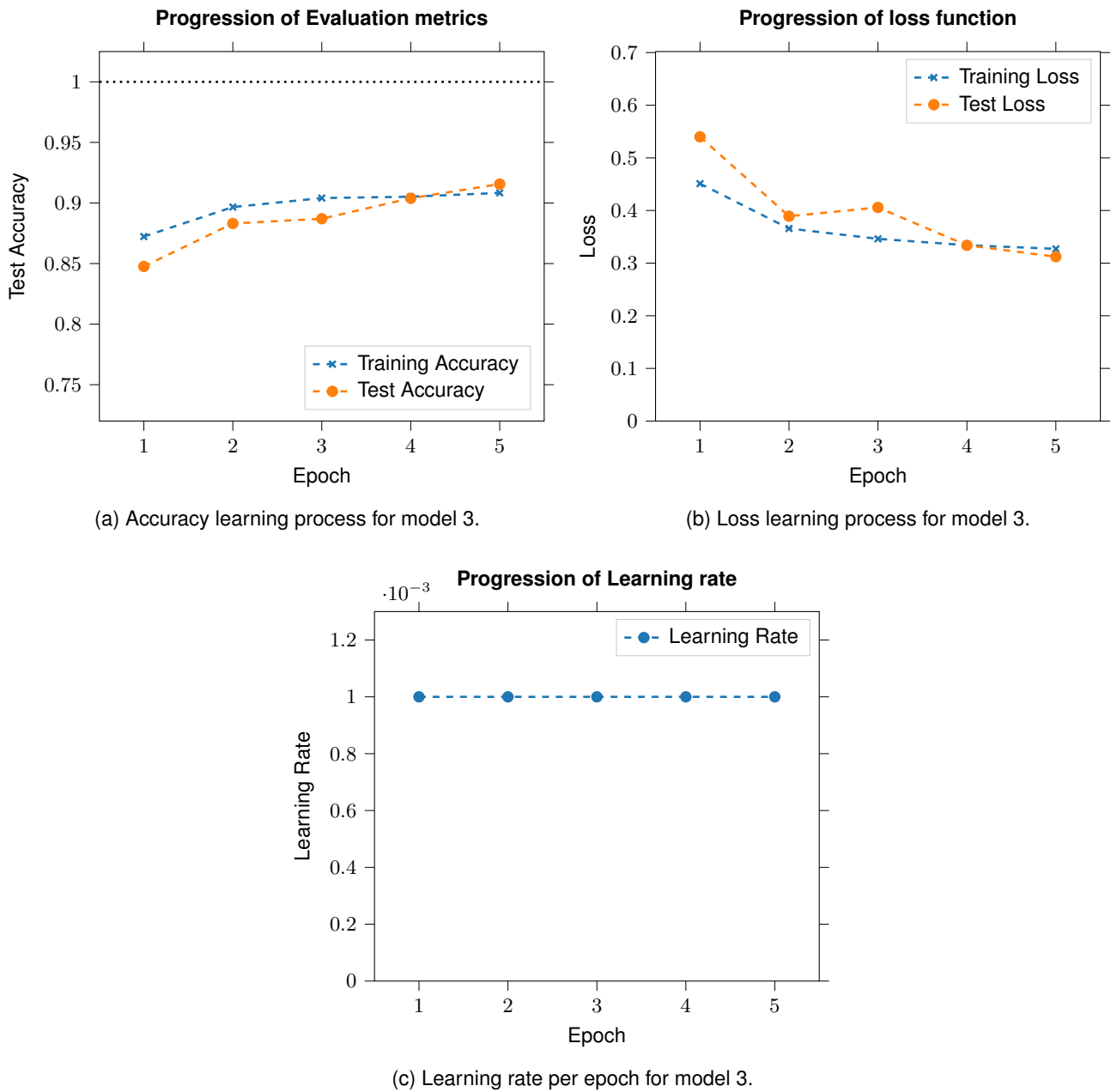


Figure 3: Training and evaluation metrics for model 3.

#### Dataset

Name MNIST

**Train-Test-Dev split:** *Training set:* 60000, *Test set:* 10000, *Dev set:* 0,

**Image size** [28, 28]

## Training

**Number of epochs** 5

**Optimizer** RMSProp (Hinton et al. 2014)

**Learning Rate** 0.0010000000474974513

**Rho** 0.8999999761581421

**Decay** 0.0

**Epsilon** 1e-07

**Loss** Categorical crossentropy

**Batch size** 64

**Shuffle** Yes

**Training time** 28 sec

## Platform

**Weights exported to path** weights\MLP2layers\_5ep\_MNIST.h5

**Device used** GPU (GeForce GTX 1060 6GB)

**CPU** Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, X86\_64

**Python Version** 3.7.2.final.0 (64 bit)

**Keras Version** 2.2.5 (Backend: tensorflow)

**Tensorflow Version** 1.14.0

**Timestamp** 26.09.2019 at 11:16

## 2.4 Model 4: MLP2layers

Training history See Figure 4.

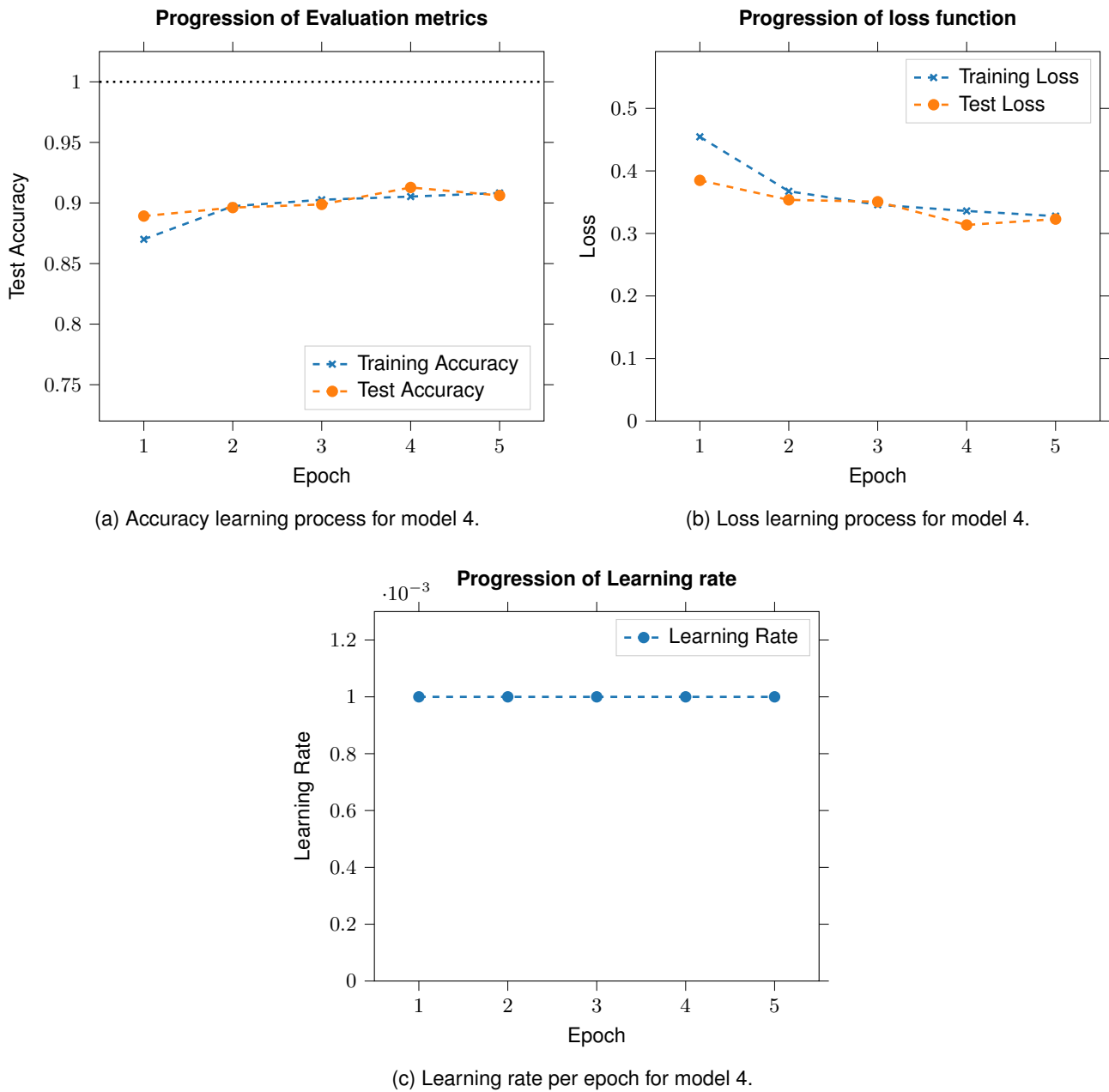


Figure 4: Training and evaluation metrics for model 4.

### Dataset

Name MNIST



**Train-Test-Dev split:** *Training set:* 60000, *Test set:* 10000, *Dev set:* 0,

**Image size** [28, 28]

## Training

**Number of epochs** 5

**Optimizer** RMSProp (Hinton et al. 2014)

**Learning Rate** 0.0010000000474974513

**Rho** 0.8999999761581421

**Decay** 0.0

**Epsilon** 1e-07

**Loss** Categorical crossentropy

**Batch size** 64

**Shuffle** Yes

**Training time** 27 sec

## Platform

**Weights exported to path** weights\MLP2layers\_5ep\_MNIST.h5

**Device used** GPU (GeForce GTX 1060 6GB)

**CPU** Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, X86\_64

**Python Version** 3.7.2.final.0 (64 bit)

**Keras Version** 2.2.5 (Backend: tensorflow)

**Tensorflow Version** 1.14.0

**Timestamp** 26.09.2019 at 11:18

## 2.5 Model 5: MLP5layers

Training history See Figure 5.

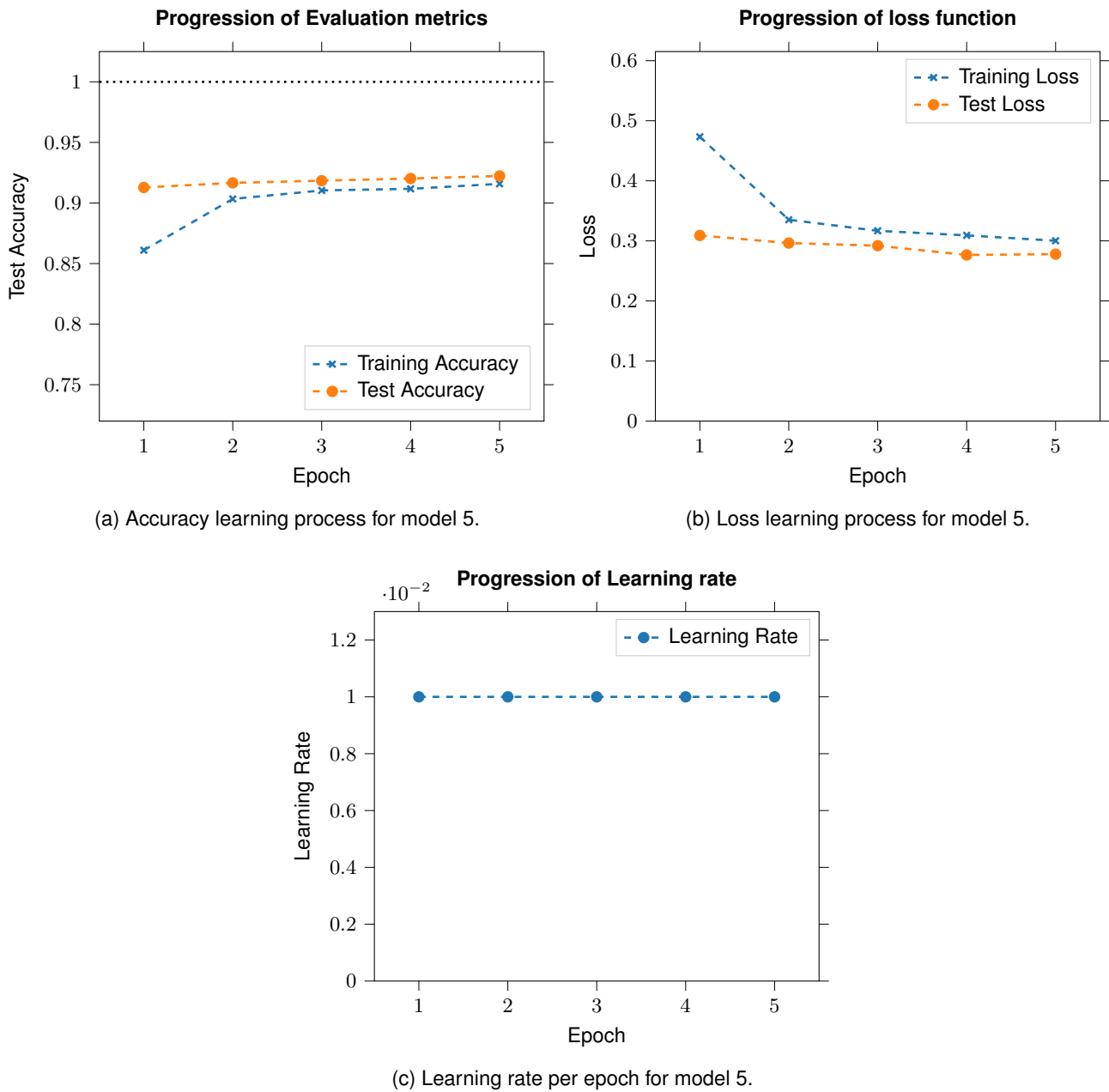


Figure 5: Training and evaluation metrics for model 5.

**Dataset**

**Name** MNIST

**Train-Test-Dev split:** *Training set:* 60000, *Test set:* 10000, *Dev set:* 0,

**Image size** [28, 28]

## Training

**Number of epochs** 5

**Optimizer** Stochastic Gradient Descent

**Learning Rate** 0.009999999776482582

**Momentum** 0.0

**Decay** 0.0

**Nesterov** False

**Loss** Categorical crossentropy

**Batch size** 64

**Shuffle** Yes

**Training time** 32 sec

## Platform

**Weights exported to path** weights\MLP5layers\_5ep\_MNIST.h5

**Device used** GPU (GeForce GTX 1060 6GB)

**CPU** Intel(R) Xeon(R) CPU E3-1245 v5 @ 3.50GHz, X86\_64

**Python Version** 3.7.2.final.0 (64 bit)

**Keras Version** 2.2.5 (Backend: tensorflow)

**Tensorflow Version** 1.14.0

**Timestamp** 26.09.2019 at 11:18

### 3 Model Architectures

#### 3.1 MLP2layers

Used in №: 3, 5

Model summary:

Nº	Layer (Type)	Output shape	Config	#Parameters	Inbound layers
0	<a href="#">input_2</a> (InputLayer)	(28, 28, 1)		0	
1	<a href="#">flatten_2</a> (Flatten)	(784,)		0	<a href="#">input_2</a>
2	<a href="#">dense_3</a> (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	401 920	<a href="#">flatten_2</a>
3	<a href="#">dropout_3</a> (Dropout)	(512,)	<b>Dropout Rate:</b> 0.0	0	<a href="#">dense_3</a>
4	<a href="#">dense_4</a> (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	262 656	<a href="#">dropout_3</a>
5	<a href="#">dropout_4</a> (Dropout)	(512,)	<b>Dropout Rate:</b> 0.2	0	<a href="#">dense_4</a>
6	<a href="#">dense_5</a> (Dense)	(10,)	<b>#Neurons:</b> 10 <b>Activation:</b> softmax	5130	<a href="#">dropout_4</a>

## 3.2 MLP5layers

Used in Nº: 5

Model summary:

Nº	Layer (Type)	Output shape	Config	#Parameters	Inbound layers
0	input_3 (InputLayer)	(28, 28, 1)		0	
1	flatten_3 (Flatten)	(784,)		0	input_3
2	dense_6 (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	401 920	flatten_3
3	dropout_5 (Dropout)	(512,)	<b>Dropout Rate:</b> 0.0	0	dense_6
4	dense_7 (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	262 656	dropout_5
5	dropout_6 (Dropout)	(512,)	<b>Dropout Rate:</b> 0.0	0	dense_7
6	dense_8 (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	262 656	dropout_6
7	dropout_7 (Dropout)	(512,)	<b>Dropout Rate:</b> 0.0	0	dense_8
8	dense_9 (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	262 656	dropout_7
9	dropout_8 (Dropout)	(512,)	<b>Dropout Rate:</b> 0.2	0	dense_9
10	dense_10 (Dense)	(512,)	<b>#Neurons:</b> 512 <b>Activation:</b> linear	262 656	dropout_8
11	dropout_9 (Dropout)	(512,)	<b>Dropout Rate:</b> 0.2	0	dense_10
12	dense_11 (Dense)	(10,)	<b>#Neurons:</b> 10 <b>Activation:</b> softmax	5130	dropout_9

### 3.3 ConvNet2layers

Used in №: 1, 5

Model summary:

Nº	Layer (Type)	Output shape	Config	#Parameters	Inbound layers
0	<a href="#">input_1</a> (InputLayer)	(28, 28, 1)		0	
1	<a href="#">conv2d_1</a> (Conv2D)	(26, 26, 32)	<b>Activation:</b> relu <b>Kernel Size:</b> [3, 3] <b>Stride:</b> [1, 1] <b>Dilation:</b> [1, 1] <b>Padding:</b> valid	320	<a href="#">input_1</a>
2	<a href="#">conv2d_2</a> (Conv2D)	(24, 24, 64)	<b>Activation:</b> relu <b>Kernel Size:</b> [3, 3] <b>Stride:</b> [1, 1] <b>Dilation:</b> [1, 1] <b>Padding:</b> valid	18 496	<a href="#">conv2d_1</a>
3	<a href="#">max_pooling2d_1</a> (MaxPooling2D)	(12, 12, 64)	<b>Pool size:</b> [2, 2] <b>Strides:</b> [2, 2] <b>Padding:</b> valid	0	<a href="#">conv2d_2</a>
4	<a href="#">dropout_1</a> (Dropout)	(12, 12, 64)	<b>Dropout Rate:</b> 0.0	0	<a href="#">max_pooling2d_1</a>
5	<a href="#">flatten_1</a> (Flatten)	(9216,)		0	<a href="#">dropout_1</a>
6	<a href="#">dense_1</a> (Dense)	(128,)	<b>#Neurons:</b> 128 <b>Activation:</b> relu	1 179 776	<a href="#">flatten_1</a>
7	<a href="#">dropout_2</a> (Dropout)	(128,)	<b>Dropout Rate:</b> 0.2	0	<a href="#">dense_1</a>
8	<a href="#">dense_2</a> (Dense)	(10,)	<b>#Neurons:</b> 10 <b>Activation:</b> softmax	1290	<a href="#">dropout_2</a>