Plot_Graphs

March 28, 2020

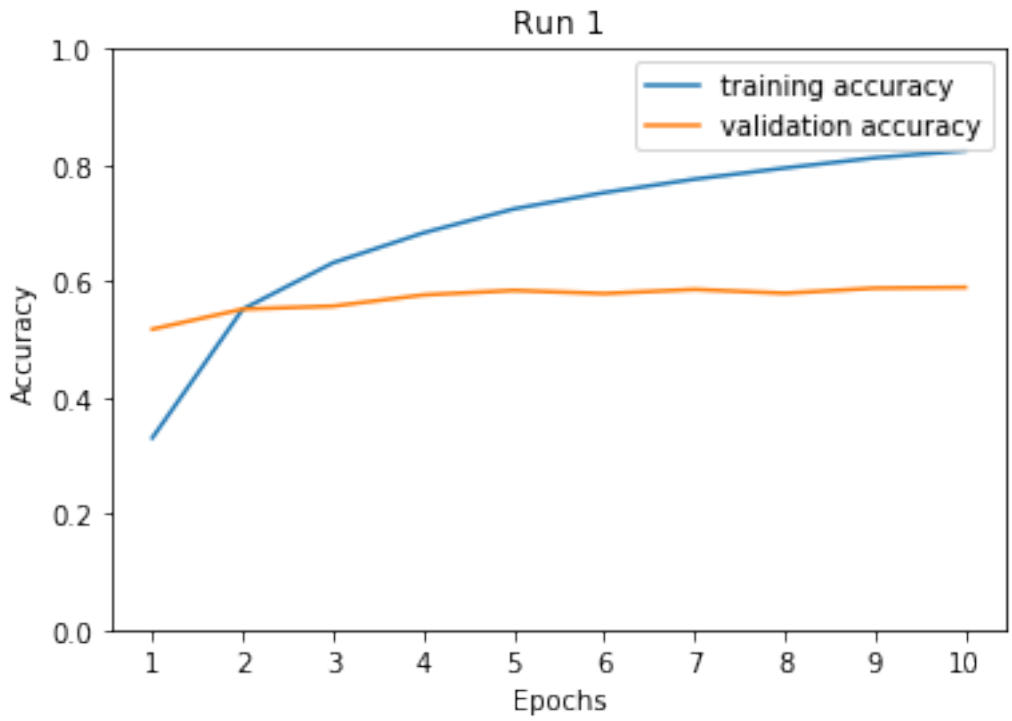
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import ylim, xticks, savefig
from matplotlib.ticker import MaxNLocator
```

[2]:
```
model_history1 = np.load('History/Run1/model_history.npy', allow_pickle=True)
```

[3]:
```
val_acc1 = model_history1.item().get('val_accuracy')
```

[4]:
```
acc1 = model_history1.item().get('accuracy')
```

[5]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = acc1
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5,6,7,8,9,10]
y2 = val_acc1
ylim((0,1))
xticks(np.arange(0,11,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 1')
plt.legend()
plt.savefig('Plots/acc_1.png')
plt.show()
```

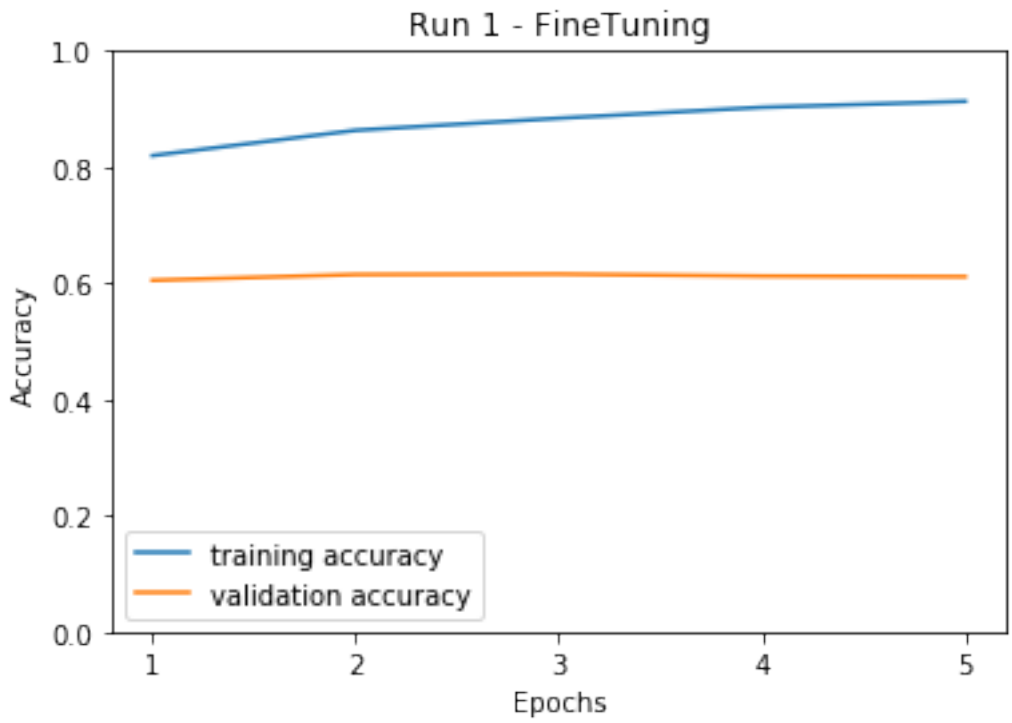


[6]:
```
ft_model_history1 = np.load('History/Run1/ft_model_history.npy',␣
↪allow_pickle=True)
```

[7]:
```
ft_val_acc1 = ft_model_history1.item().get('val_accuracy')
```

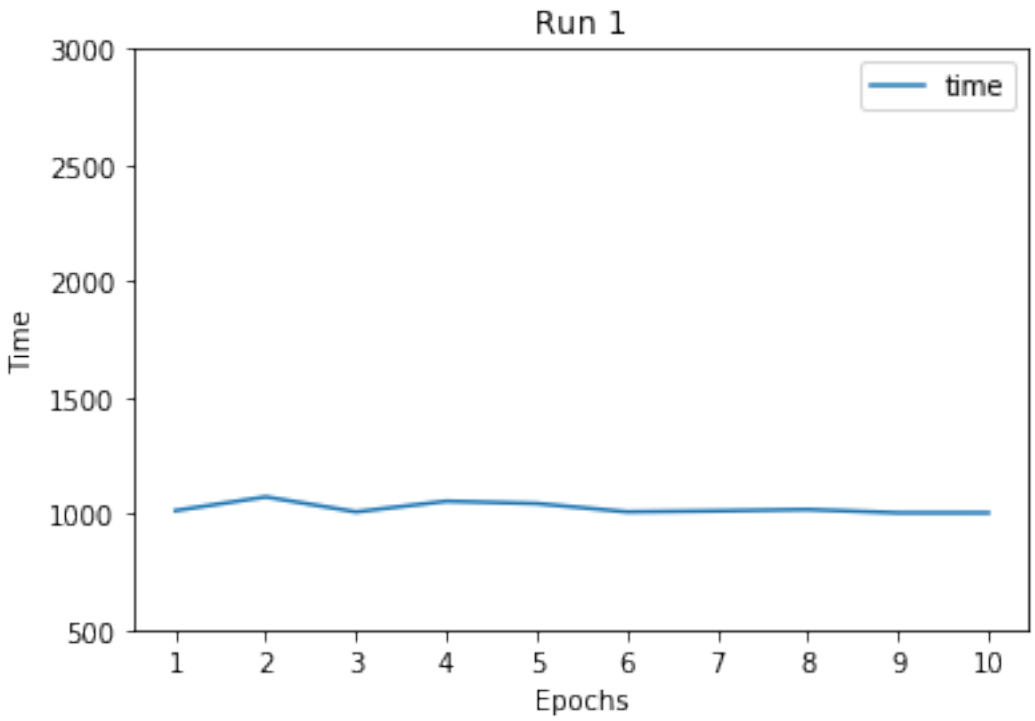
[8]:
```
ft_acc1 = ft_model_history1.item().get('accuracy')
```

[9]:
```
x1 = [1,2,3,4,5]
y1 = ft_acc1
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5]
y2 = ft_val_acc1
ylim((0,1))
xticks(np.arange(0,6,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 1 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_acc_1.png')
plt.show()
```



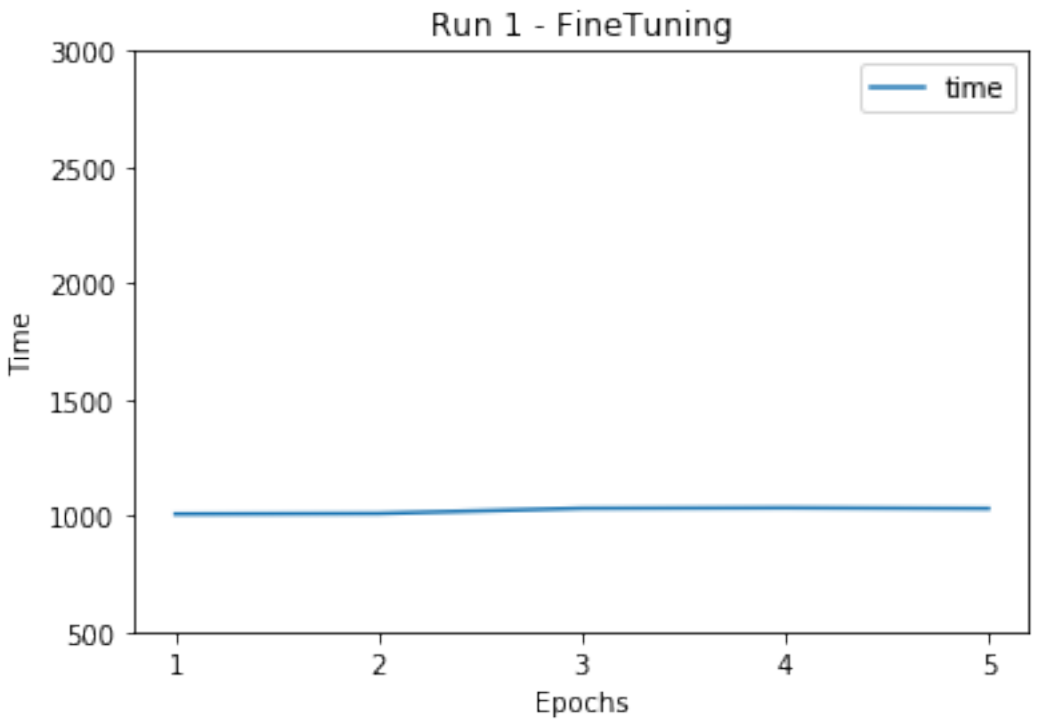
[10]:
```
model_time1 = np.load('History/Run1/model_time.npy')
```

[11]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = model_time1
ylim((500,3000))
xticks(np.arange(0,11,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 1')
plt.legend()
plt.savefig('Plots/time_1.png')
plt.show()
```



[12]:
```
ft_model_time1 = np.load('History/Run1/ft_model_time.npy')
```

[13]:
```
x1 = [1,2,3,4,5]
y1 = ft_model_time1
ylim((500,3000))
xticks(np.arange(0,6,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 1 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_time_1.png')
plt.show()
```

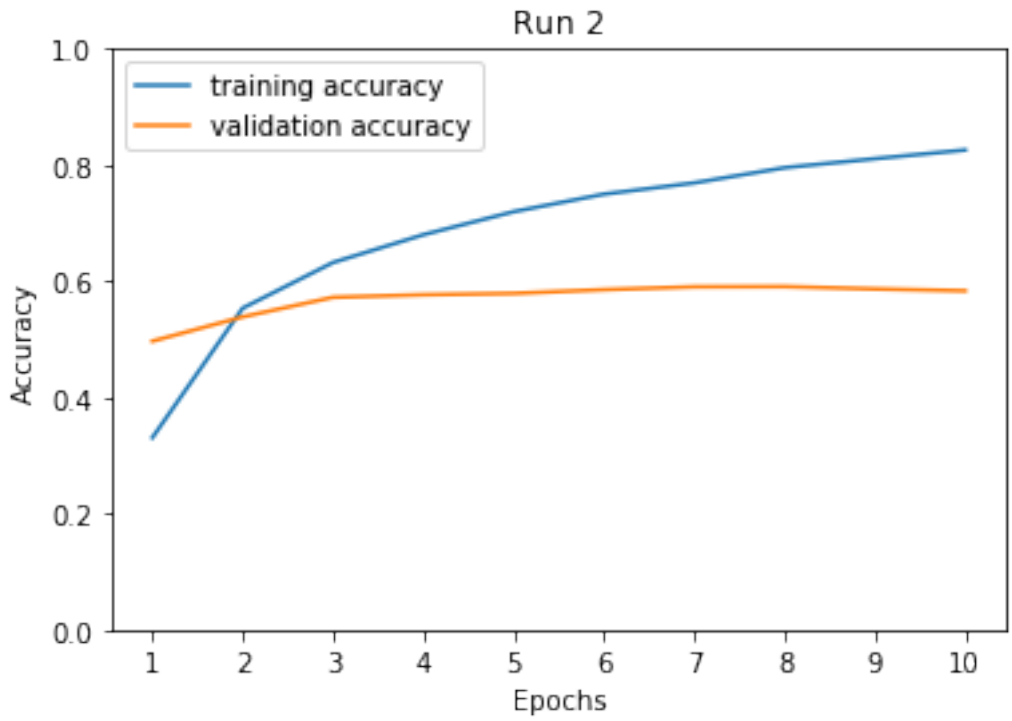


[14]:
```
model_history2 = np.load('History/Run2/model_history.npy', allow_pickle=True)
```

[15]:
```
val_acc2 = model_history2.item().get('val_accuracy')
```

[16]:
```
acc2 = model_history2.item().get('accuracy')
```

[17]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = acc2
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5,6,7,8,9,10]
y2 = val_acc2
ylim((0,1))
xticks(np.arange(0,11,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 2')
plt.legend()
plt.savefig('Plots/acc_2.png')
plt.show()
```

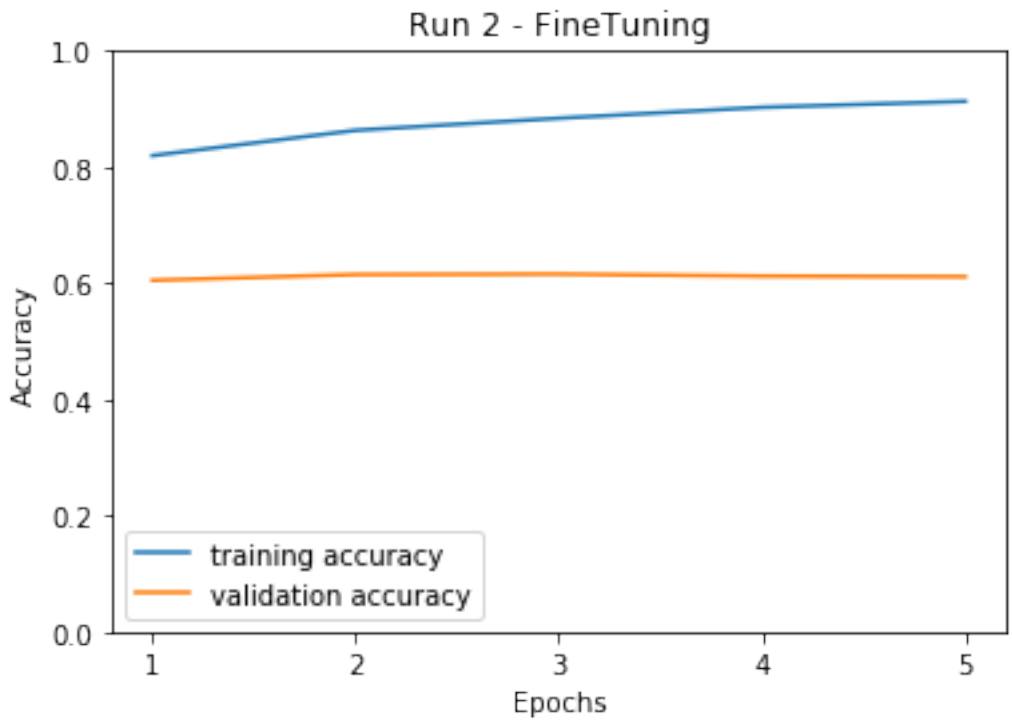


[18]:
```
ft_model_history2 = np.load('History/Run1/ft_model_history.npy',␣
↪allow_pickle=True)
```

[19]:
```
ft_val_acc2 = ft_model_history2.item().get('val_accuracy')
```

[20]:
```
ft_acc2 = ft_model_history2.item().get('accuracy')
```

[21]:
```
x1 = [1,2,3,4,5]
y1 = ft_acc2
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5]
y2 = ft_val_acc2
ylim((0,1))
xticks(np.arange(0,6,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 2 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_acc_2.png')
plt.show()
```

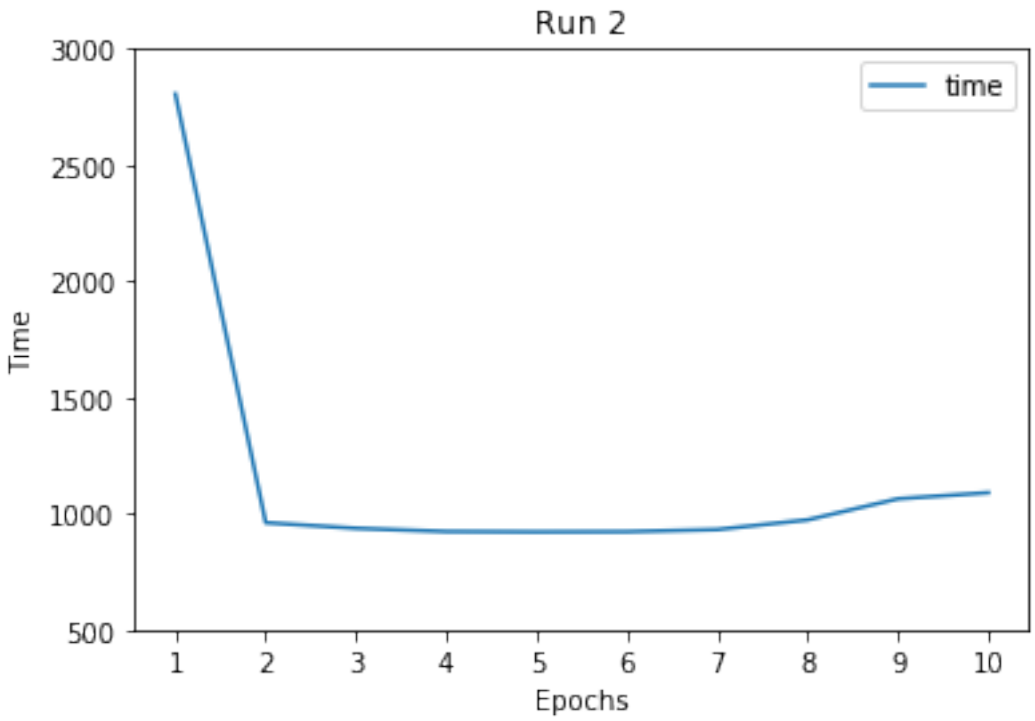


[22]:
```
model_time2 = np.load('History/Run2/model_time.npy')
```

[23]:
```
model_time2
```

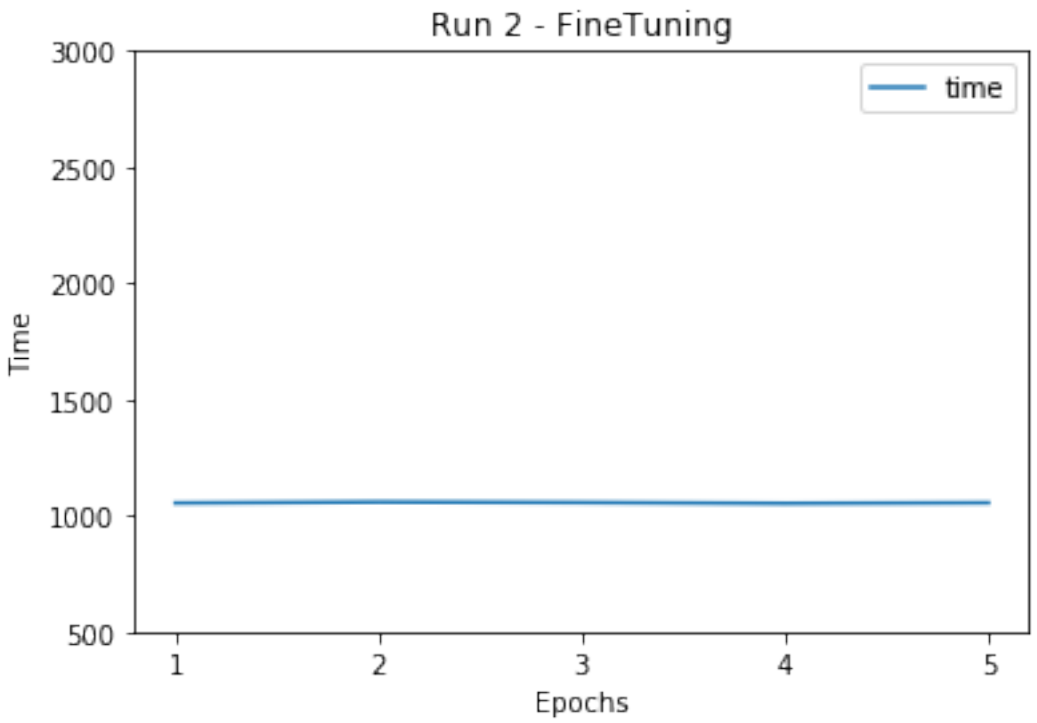
[23]:
```
array([2805.47685719,  961.16021466,  936.914047  ,  923.19990373,
        921.78762484,  922.58433151,  931.73474789,  973.32667923,
       1063.9994061 , 1090.18503141])
```

[24]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = model_time2
ylim((500,3000))
xticks(np.arange(0,11,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 2')
plt.legend()
plt.savefig('Plots/time_2.png')
plt.show()
```



[25]:
```
ft_model_time2 = np.load('History/Run2/ft_model_time.npy')
```

[26]:
```
x1 = [1,2,3,4,5]
y1 = ft_model_time2
ylim((500,3000))
xticks(np.arange(0,6,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 2 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_time_2.png')
plt.show()
```

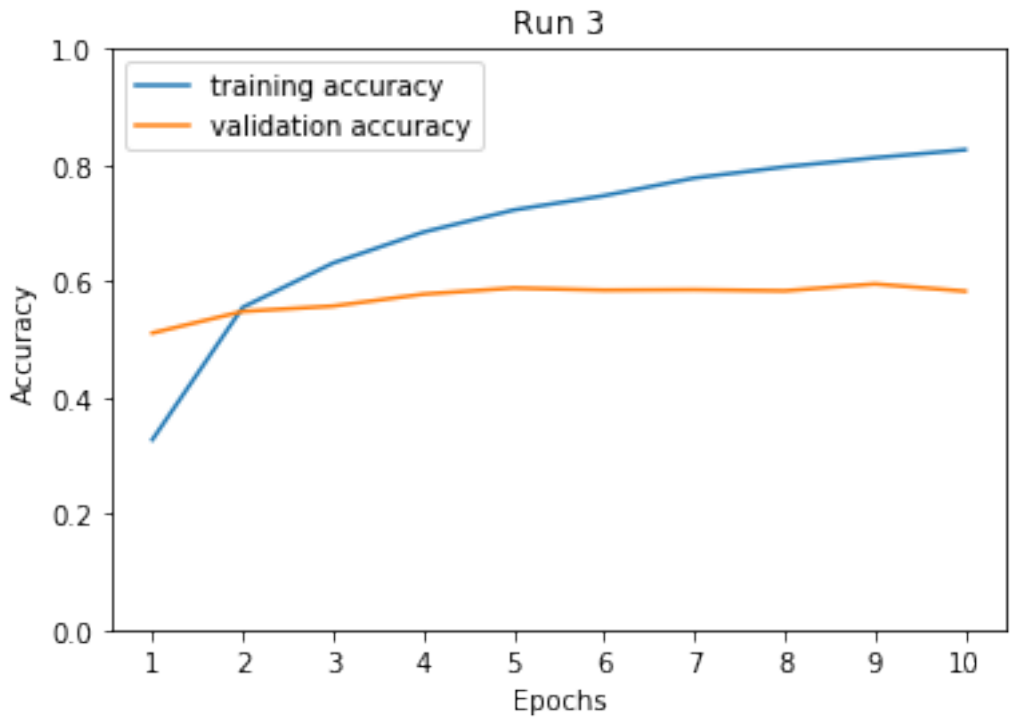


[27]:
```
model_history3 = np.load('History/Run3/model_history.npy', allow_pickle=True)
```

[28]:
```
val_acc3 = model_history3.item().get('val_accuracy')
```

[29]:
```
acc3 = model_history3.item().get('accuracy')
```

[30]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = acc3
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5,6,7,8,9,10]
y2 = val_acc3
ylim((0,1))
xticks(np.arange(0,11,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 3')
plt.legend()
plt.savefig('Plots/acc_3.png')
plt.show()
```

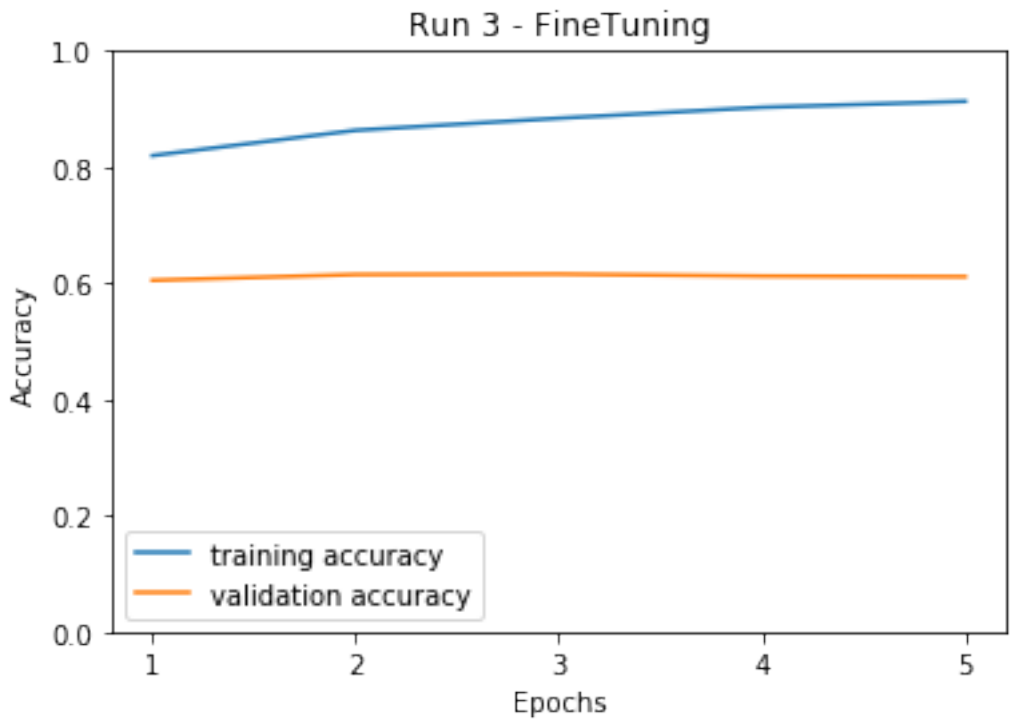


[31]:
```
ft_model_history3 = np.load('History/Run1/ft_model_history.npy',␣
↪allow_pickle=True)
```

[32]:
```
ft_val_acc3 = ft_model_history3.item().get('val_accuracy')
```

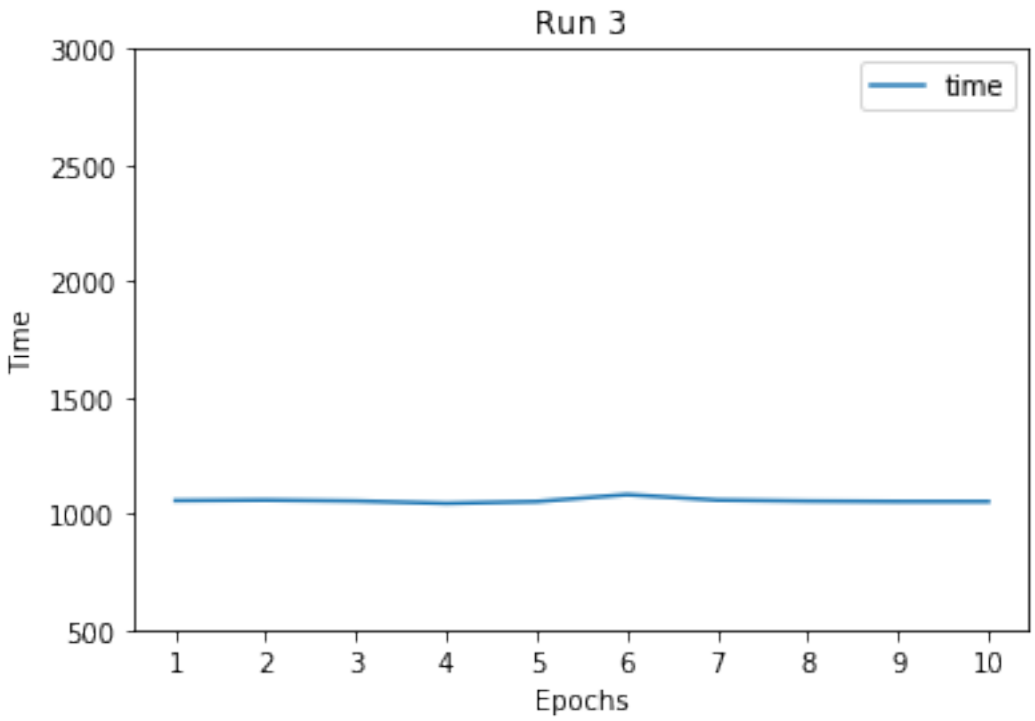
[33]:
```
ft_acc3 = ft_model_history3.item().get('accuracy')
```

[34]:
```
x1 = [1,2,3,4,5]
y1 = ft_acc3
plt.plot(x1, y1, label = "training accuracy")
x2 = [1,2,3,4,5]
y2 = ft_val_acc3
ylim((0,1))
xticks(np.arange(0,6,step=1))
plt.plot(x2, y2, label = "validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Run 3 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_acc_3.png')
plt.show()
```



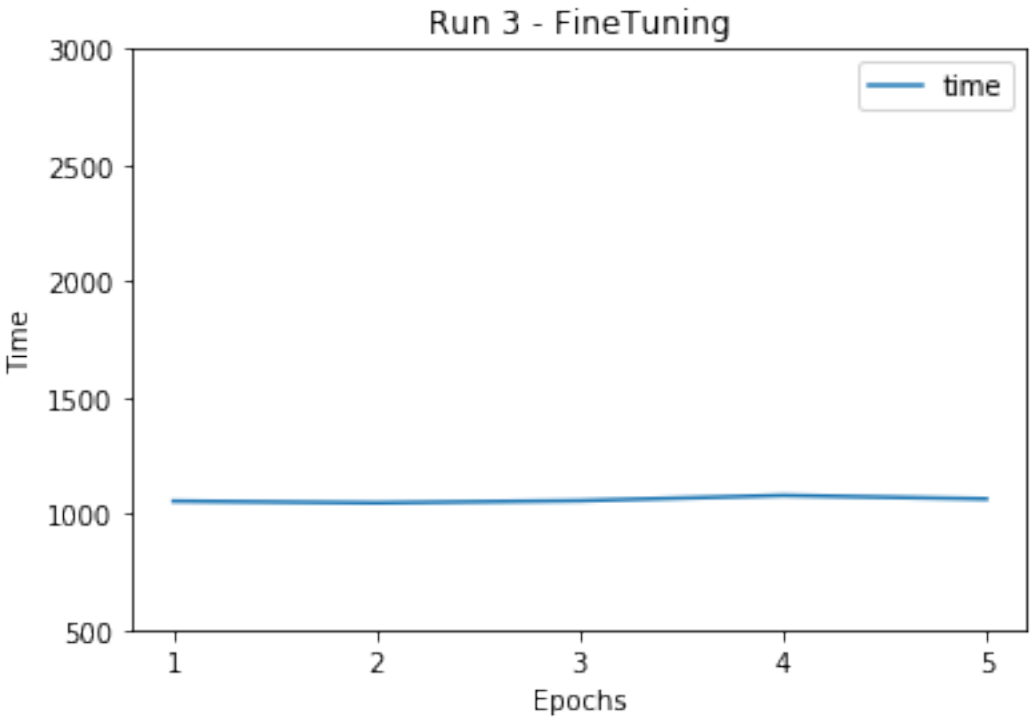
[35]:
```
model_time3 = np.load('History/Run3/model_time.npy')
```

[36]:
```
x1 = [1,2,3,4,5,6,7,8,9,10]
y1 = model_time3
ylim((500,3000))
xticks(np.arange(0,11,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 3')
plt.legend()
plt.savefig('Plots/time_3.png')
plt.show()
```



[37]:
```
ft_model_time3 = np.load('History/Run3/ft_model_time.npy')
```

[38]:
```
x1 = [1,2,3,4,5]
y1 = ft_model_time3
ylim((500,3000))
xticks(np.arange(0,6,step=1))
plt.plot(x1, y1, label = "time")
plt.xlabel('Epochs')
plt.ylabel('Time')
plt.title('Run 3 - FineTuning')
plt.legend()
plt.savefig('Plots/ft_time_3.png')
plt.show()
```


Run 3 - FineTuning
time
Time
Epochs
3000
2500
2000
1500
1000
500
1
2
3
4
5

[ ]: