Project music genres classification

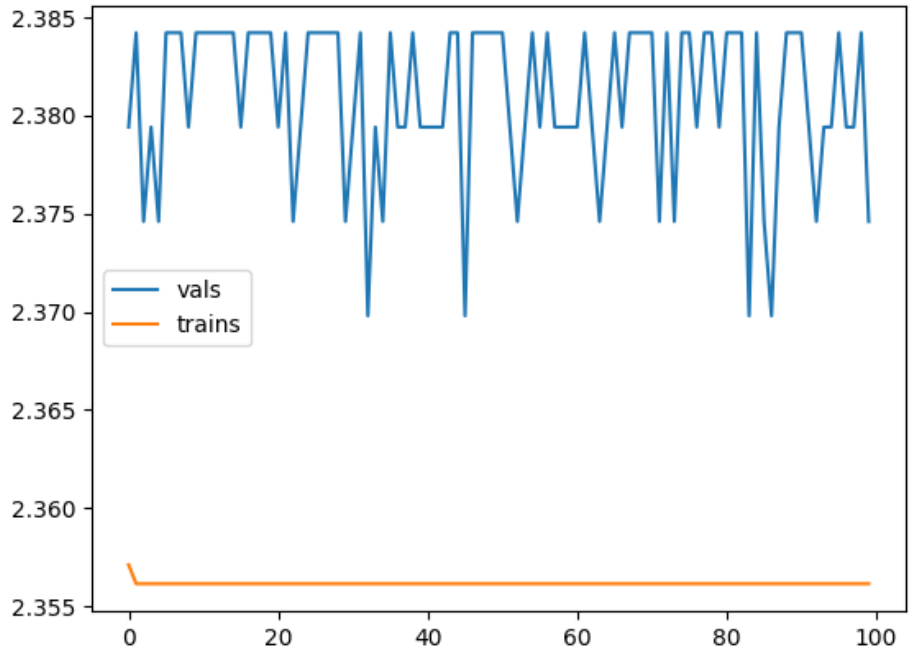
Model

```python
import torch.nn as nn
import torch.nn.functional as F

class ClassifierMusicGenres(nn.Module):
    def __init__(self, inputSize, numclasses):
        super(ClassifierMusicGenres,self).__init__()
        self.inputSize = inputSize
        self.fc1=nn.Linear(inputSize,512)
        self.fc2=nn.Linear(512,256)
        self.fc3=nn.Linear(256,64)
        self.fc4=nn.Linear(64,numclasses)

    def forward(self,x):
        #x = x.view(x.size(0), -1)
        x=F.relu(self.fc1(x))
        x=F.relu(self.fc2(x))
        x=F.relu(self.fc3(x))
        x=F.softmax(self.fc4(x), dim=1)
        return x
```

Config

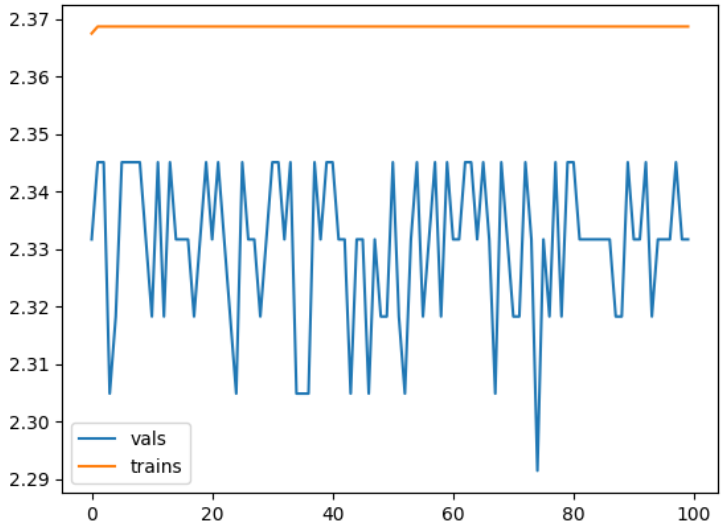
```
input_size: 26406
num_classes: 10
learning_rate: .001
batch_size: 16
num_epochs: 100
```

Output



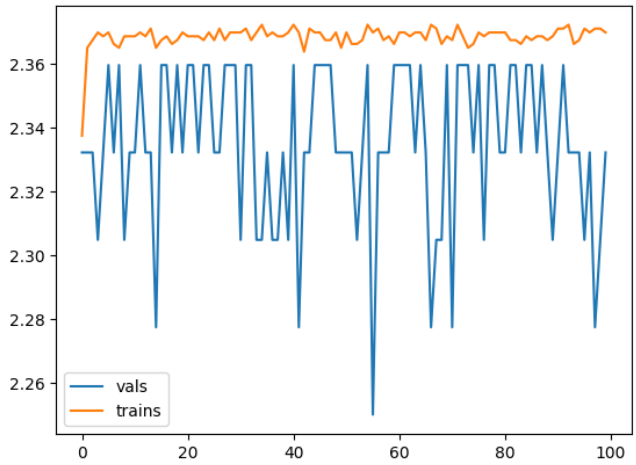
Config – changed batch size to 32:

```
input_size: 26406
num_classes: 10
learning_rate: .001
batch_size: 32
num_epochs: 100
```



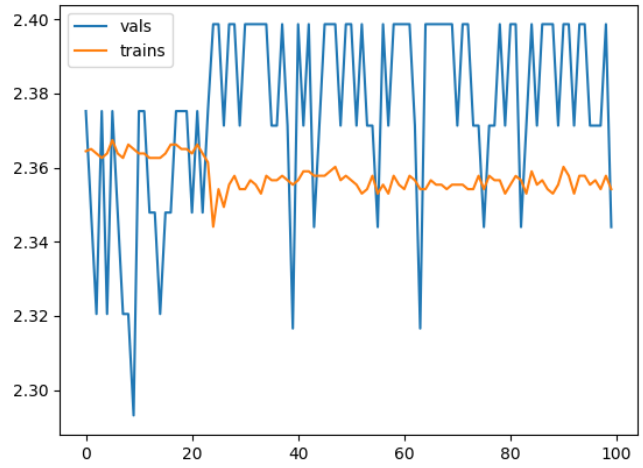
Config – changed batch size to 64:

```
input_size: 26406
num_classes: 10
learning_rate: .001
batch_size: 64
num_epochs: 100
```



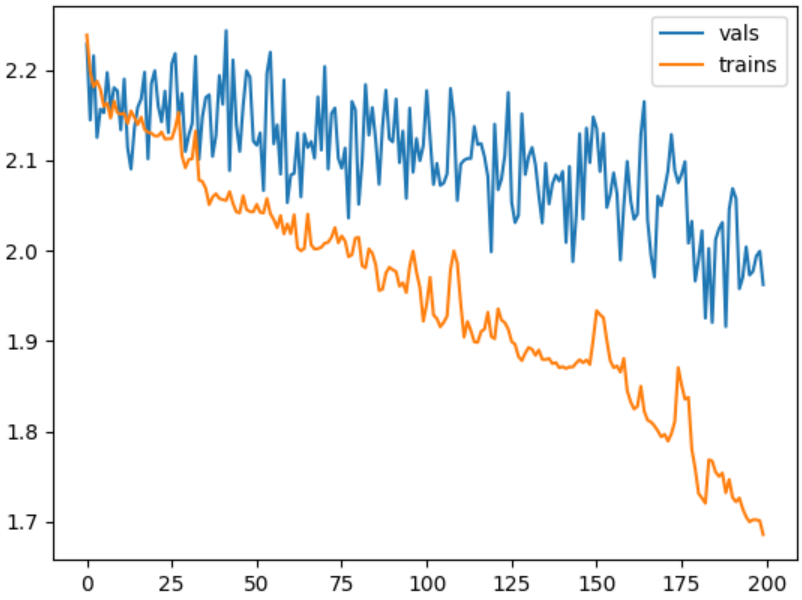
Config – changed learning rate to .0001:

```
input_size: 26406
num_classes: 10
learning_rate: .0001
batch_size: 64
num_epochs: 100
```



Config – changed num epochs to 200

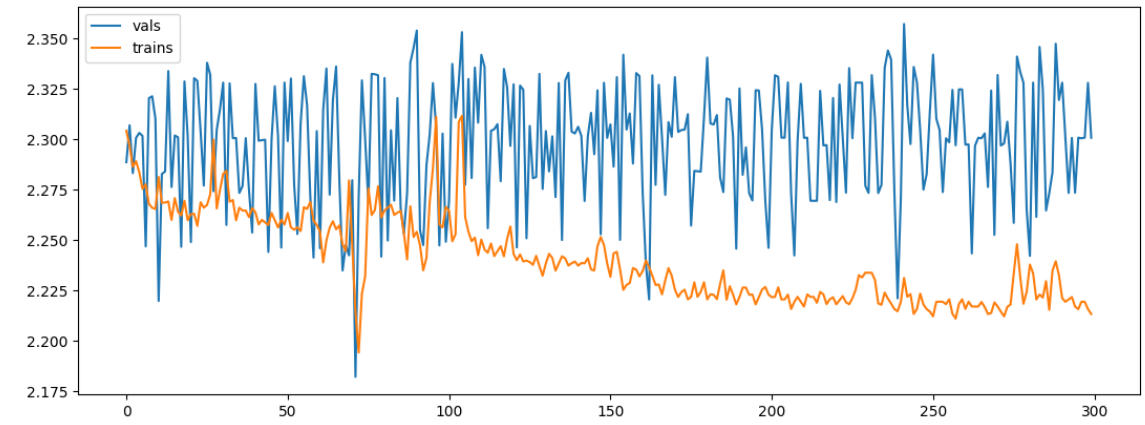
```
learning_rate: .0001
batch_size: 64
num_epochs: 200
```

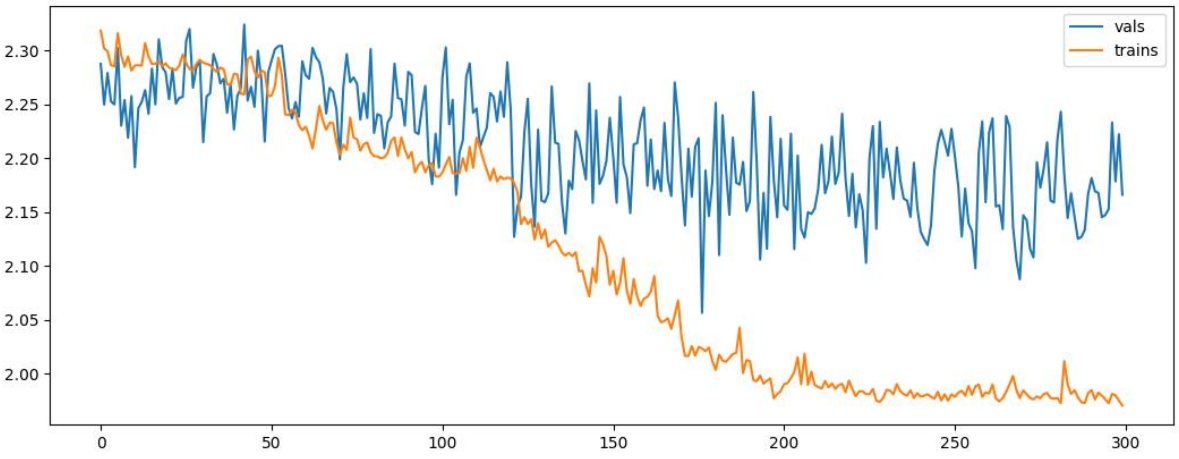


Config – changed num epochs to 300:

```
learning_rate: .0001
batch_size: 64
num_epochs: 300
```

Different runs:



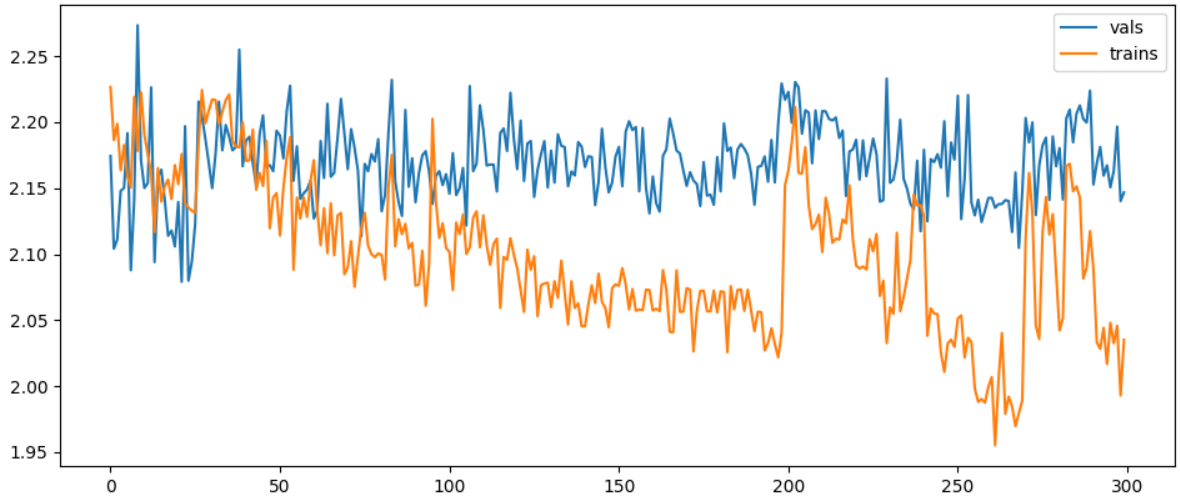


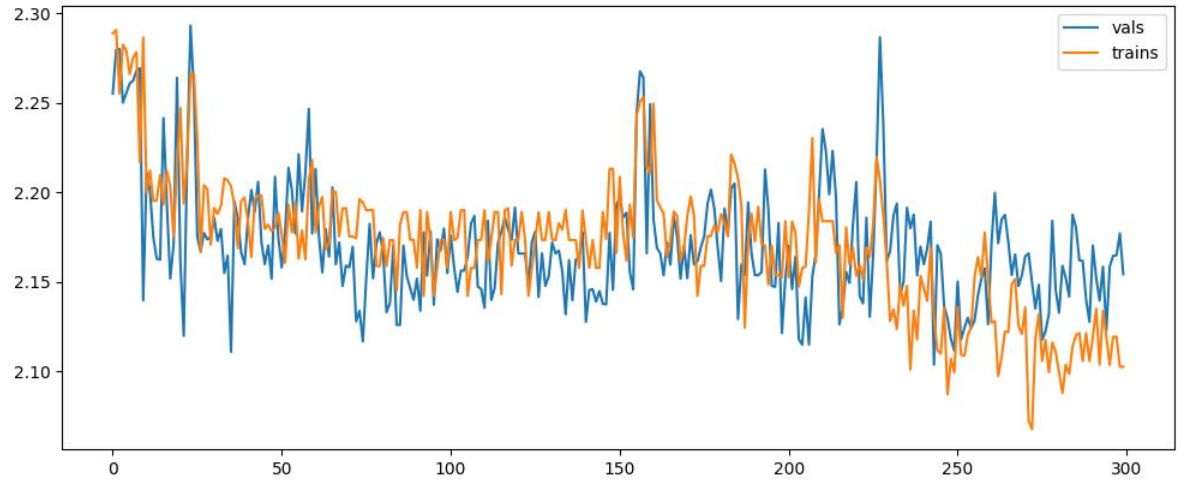
Config remains, changes train/val data ratio

```
len1 = int(len(dataset)*.9)
len2 = int(len(dataset)*.1)
```

(Was .8/.2)

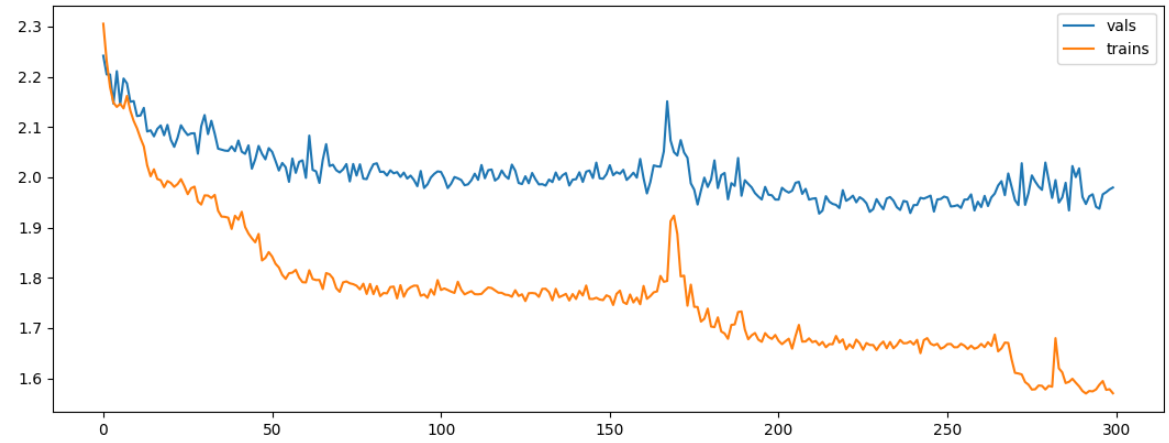
Different runs





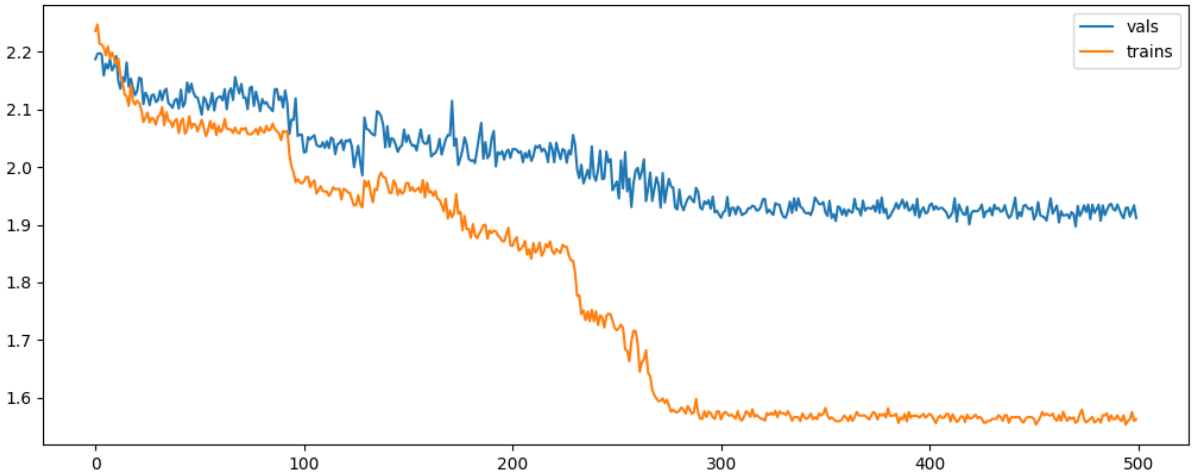
Back to .8/.2 ratio and batch size to 128

```
learning_rate: .0001
batch_size: 128
num_epochs: 300
```



Config num epochs to 500:

```
learning_rate: .0001
batch_size: 128
num_epochs: 500
```



//testing 300+ epochs makes little change = not much sense then here

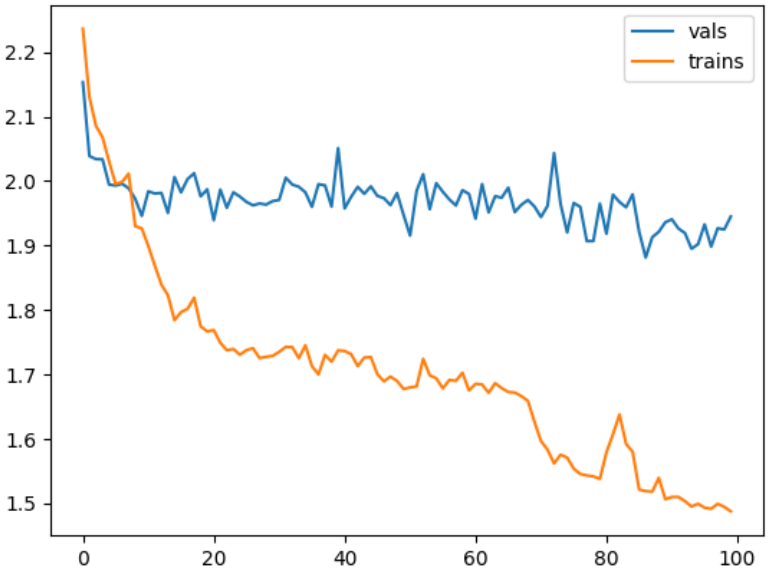
New model:

```python
class ClassifierMusicGenres(nn.Module):
    def __init__(self, inputSize, numclasses):
        super(ClassifierMusicGenres,self).__init__()
        self.inputSize = inputSize
        self.fc1=nn.Linear(inputSize,512)
        self.fc2=nn.Linear(512,1024)
        self.fc3=nn.Linear(1024,2048)
        self.fc4=nn.Linear(2048,512)
        self.fc5=nn.Linear(512,256)
        self.fc6=nn.Linear(256,64)
        self.fc7=nn.Linear(64,numclasses)

    def forward(self,x):
        #x = x.view(x.size(0), -1)
        x=F.relu(self.fc1(x))
        x=F.relu(self.fc2(x))
        x=F.relu(self.fc3(x))
        x=F.relu(self.fc4(x))
        x=F.relu(self.fc5(x))
        x=F.relu(self.fc6(x))
        x=F.softmax(self.fc7(x), dim=1)
        return x
```

Config:

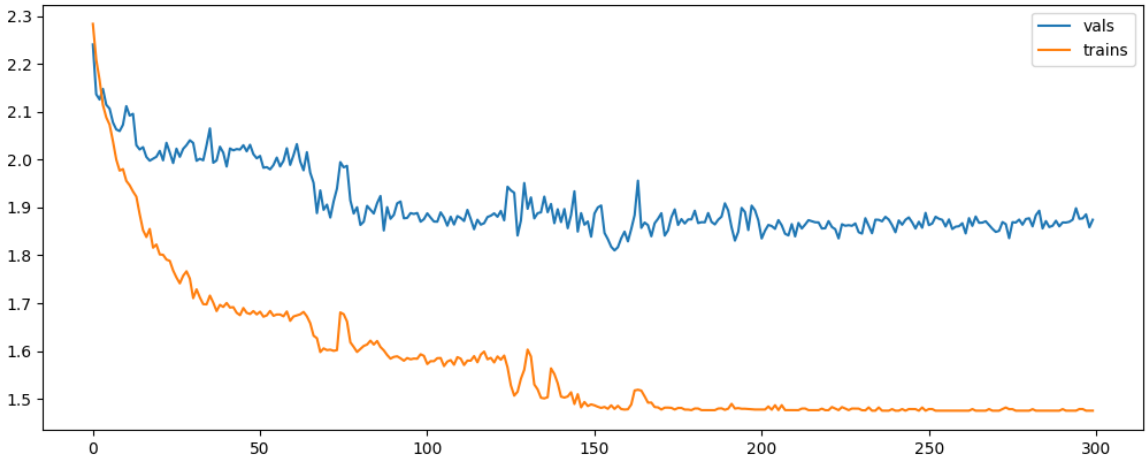
```
input_size: 26406
num_classes: 10
learning_rate: .0001
batch_size: 128
num_epochs: 100
```

Output:



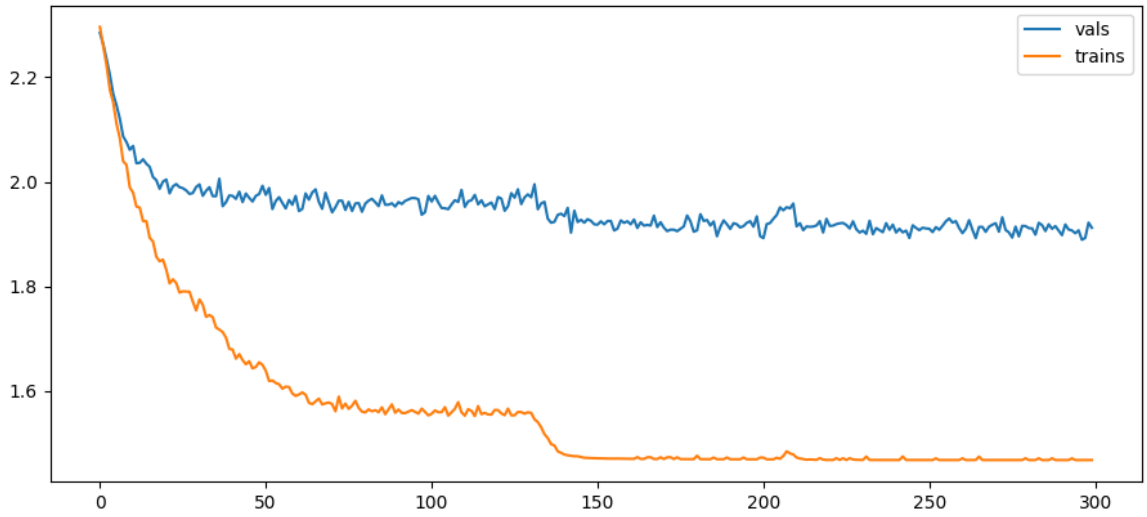
Config – changed num epochs to 300

```
learning_rate: .0001
batch_size: 128
num_epochs: 300
```



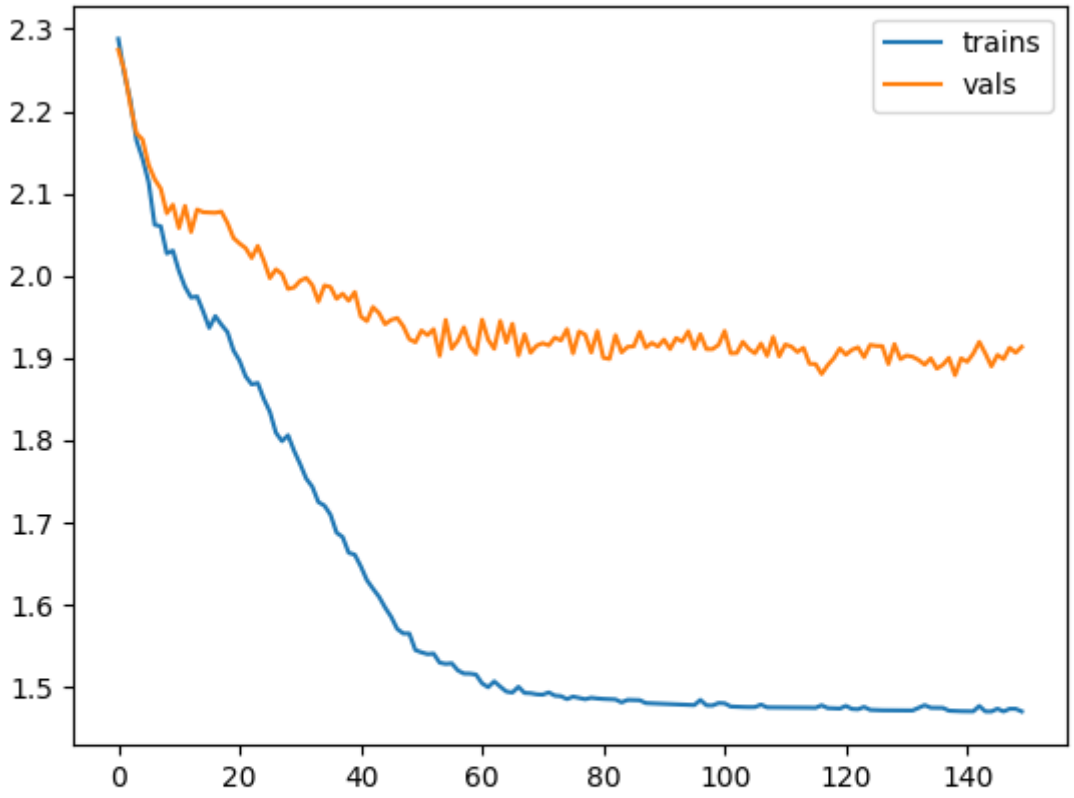
Config – changed learning rate to .00001

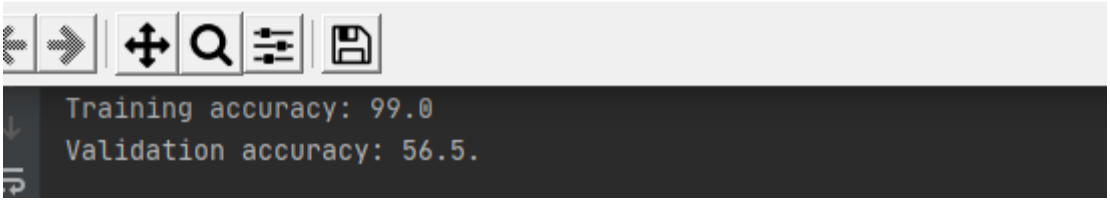
```
learning_rate: .00001
batch_size: 128
num_epochs: 300
```



After adding check_acc function (colors on plot reversed!):

```
learning_rate: .00001
batch_size: 128
num_epochs: 150
```

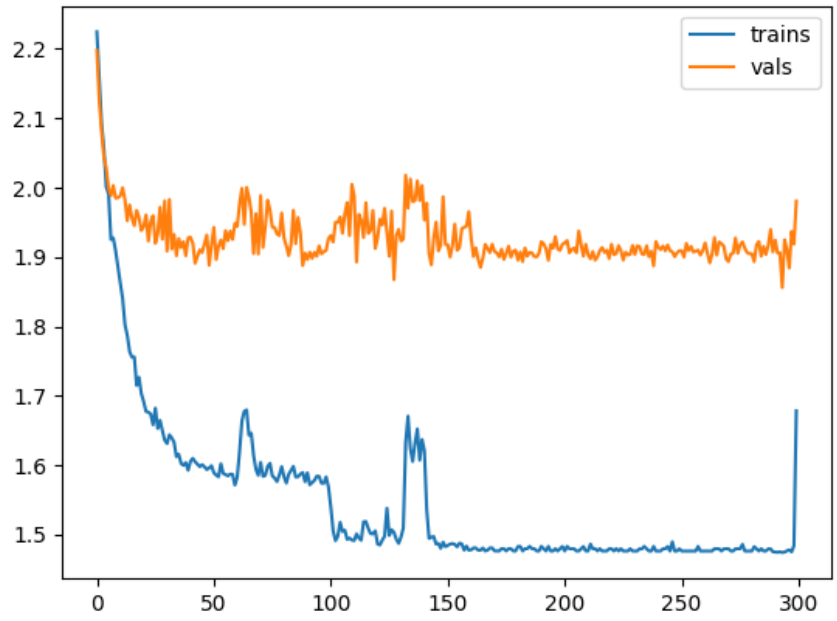




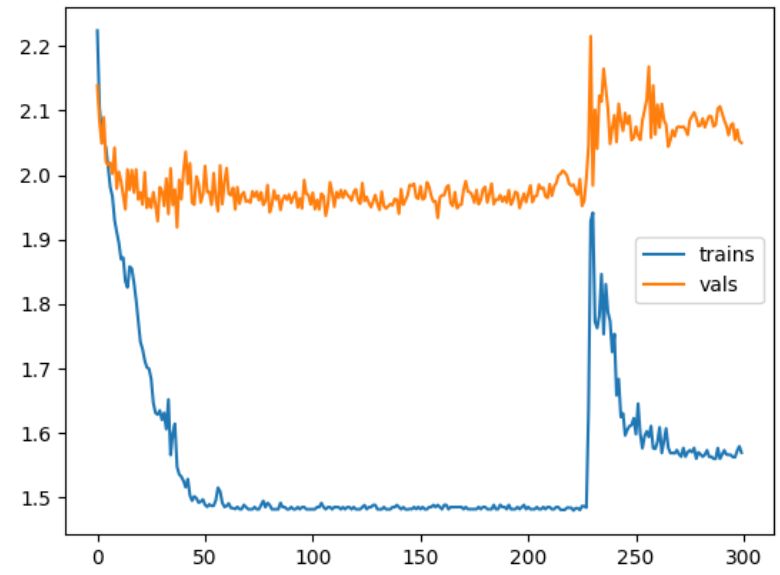
```
Training accuracy: 99.0
Validation accuracy: 56.5.
```

On lower learning rate and more num_epochs:

```
learning_rate: .0001
batch_size: 128
num_epochs: 300
```




```
Training accuracy: 65.62
Validation accuracy: 47.5.
```

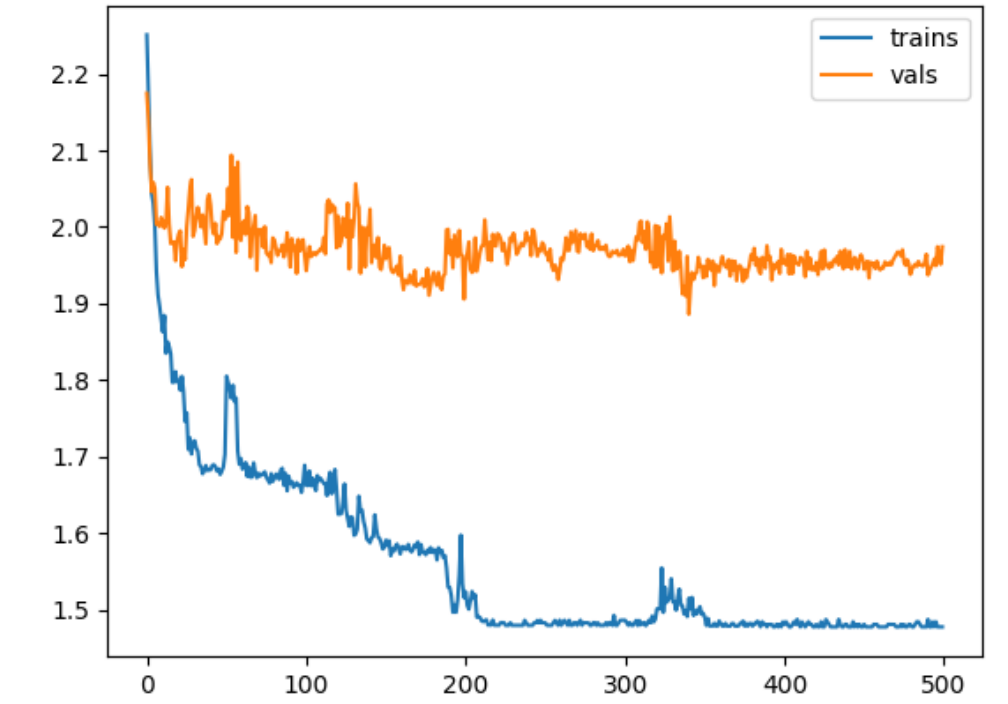



```
Training accuracy: 89.38
Validation accuracy: 40.0.
```

\\ What kind of magic is that

Config – changed to num_epochs = 500 to actually see what could come next

```
learning_rate: .0001
batch_size: 128
num_epochs: 500
```




```
Training accuracy: 98.12
Validation accuracy: 50.0.
```

// Fair enough