Default: Optimizer = “SDG” Loss = “Binary Cross Entropy”

jupyter Modify Parameters_ Shruti Last Checkpoint: 6 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
[40]: model.compile(optimizer='SGD',loss='BinaryCrossentropy', metrics=['accuracy']) # compiling the model

[42]: # training the model
history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

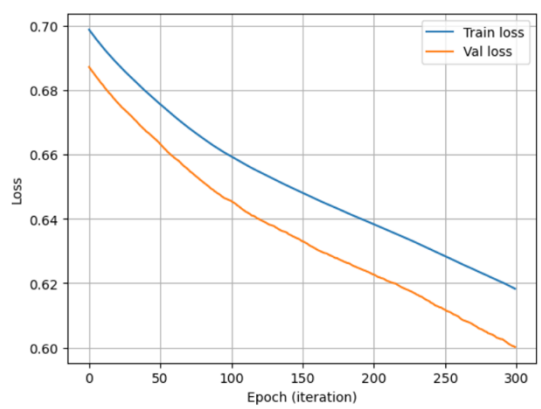
```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.5639 - loss: 0.6772 - val_accuracy: 0.2162 - val_loss: 0.6805
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.2804 - loss: 0.6771 - val_accuracy: 0.7027 - val_loss: 0.6801
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5979 - loss: 0.6768 - val_accuracy: 0.7027 - val_loss: 0.6797
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.6172 - loss: 0.6765 - val_accuracy: 0.7027 - val_loss: 0.6793
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.6120 - loss: 0.6756 - val_accuracy: 0.7027 - val_loss: 0.6788
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.6172 - loss: 0.6761 - val_accuracy: 0.7027 - val_loss: 0.6785
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.6068 - loss: 0.6749 - val_accuracy: 0.7027 - val_loss: 0.6779
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.6120 - loss: 0.6750 - val_accuracy: 0.7027 - val_loss: 0.6775
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.6016 - loss: 0.6743 - val_accuracy: 0.7027 - val_loss: 0.6771
```

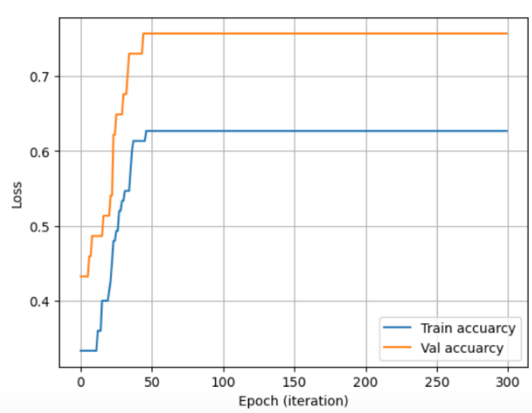
jupyter Modify Parameters_ Shruti Last Checkpoint: 6 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
[-0.32508314 -0.4028413  -0.35844663]
[-0.38873833 -0.47707173 -0.03810375]
[ 0.13462085  0.6270411   0.63946694]
[ 0.44620568 -0.49328163 -0.7331759 ]
[ 0.38265732  0.30577248 -0.20141634]
[-0.04039436 -0.37620765  0.24138576]
[-0.18675147 -0.5417032   0.18190359]
[-0.26768756 -0.59664524  0.2420639 ]]
```





1. model.compile(optimizer='SGD',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dumps

jupyter Modify Parameters_ Shruti Last Checkpoint: 7 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

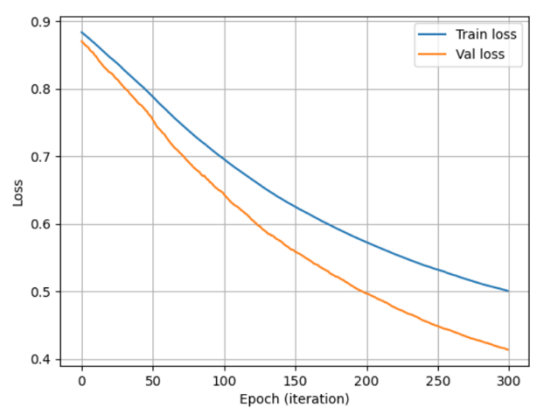
```
[43]: model.compile(optimizer='SGD',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model
      history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

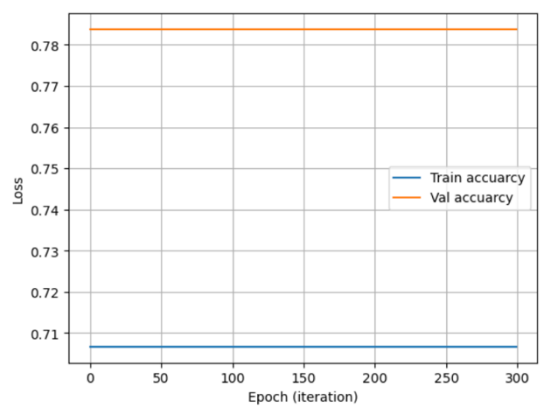
```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.7055 - loss: 0.8853 - val_accuracy: 0.7838 - val_loss: 0.8705
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7107 - loss: 0.8802 - val_accuracy: 0.7838 - val_loss: 0.8683
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.7107 - loss: 0.8766 - val_accuracy: 0.7838 - val_loss: 0.8663
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7107 - loss: 0.8768 - val_accuracy: 0.7838 - val_loss: 0.8639
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7107 - loss: 0.8770 - val_accuracy: 0.7838 - val_loss: 0.8625
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7055 - loss: 0.8743 - val_accuracy: 0.7838 - val_loss: 0.8603
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.6951 - loss: 0.8759 - val_accuracy: 0.7838 - val_loss: 0.8567
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7159 - loss: 0.8680 - val_accuracy: 0.7838 - val_loss: 0.8552
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7159 - loss: 0.8661 - val_accuracy: 0.7838 - val_loss: 0.8537
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 8 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)






```
[48]: model.compile(optimizer='SGD',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model
      history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

2. model.compile(optimizer='RMSprop'Links to an external site.",loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dumps

jupyter Modify Parameters_ Shruti Last Checkpoint: 9 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[52]:
```
model.compile(optimizer='RMSprop',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model
history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.8360 - loss: 0.3913 - val_accuracy: 0.8378 - val_loss: 0.2970
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8078 - loss: 0.3894 - val_accuracy: 0.8378 - val_loss: 0.2964
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8042 - loss: 0.3812 - val_accuracy: 0.8378 - val_loss: 0.2949
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8183 - loss: 0.3787 - val_accuracy: 0.8378 - val_loss: 0.2936
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8308 - loss: 0.3776 - val_accuracy: 0.8378 - val_loss: 0.2931
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8078 - loss: 0.3808 - val_accuracy: 0.8378 - val_loss: 0.2921
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8360 - loss: 0.3822 - val_accuracy: 0.8378 - val_loss: 0.2914
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8131 - loss: 0.3784 - val_accuracy: 0.8378 - val_loss: 0.2901
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8026 - loss: 0.3838 - val_accuracy: 0.8378 - val_loss: 0.2903
```

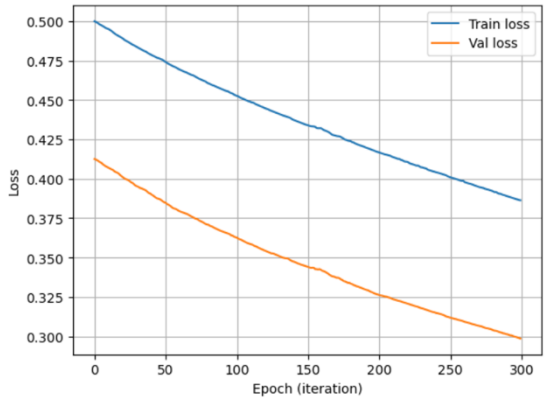
[53]:
```
loss = model.evaluate(X_test, Y_oh_test, verbose=0)
print('Test loss (cross-entropy and accuracy):',loss)
print()
W = model.get_weights()
for ii in range(len(W)//2):
    print("Layer %d" %ii)
    print('Bias:\n', W[2*ii + 1])
    print('W:\n', W[2*ii])
    print()

plt.plot(history.history['loss'], label = "Train loss")
plt.plot(history.history['val_loss'], label = "Val loss")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()


plt.plot(history.history['accuracy'], label = "Train accuarcy")
plt.plot(history.history['val_accuracy'], label = "Val accuarcy")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()
```

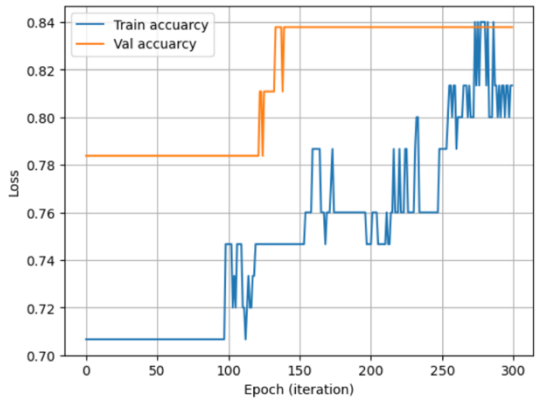
```
Test loss (cross-entropy and accuracy): [0.3185528814792633, 0.8947368264198303]
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 9 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)





3. model.compile(optimizer='adadelta' ',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dumps

jupyter Modify Parameters_ Shruti Last Checkpoint: 9 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[56]:
```
model.compile(optimizer='Adadelta',loss='categorical_crossentropy', metrics=['accuracy']) # compiling the model
history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9347 - loss: 0.2184 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9295 - loss: 0.2353 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9399 - loss: 0.2235 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9295 - loss: 0.2265 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9295 - loss: 0.2267 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.9347 - loss: 0.2207 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.2234 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9295 - loss: 0.2289 - val_accuracy: 0.9459 - val_loss: 0.1622
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.2257 - val_accuracy: 0.9459 - val_loss: 0.1622
```

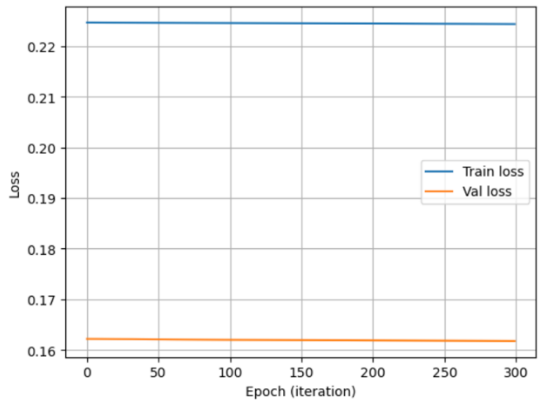
[57]:
```
loss = model.evaluate(X_test, Y_oh_test, verbose=0)
print('Test loss (cross-entropy and accuracy):',loss)
print()
W = model.get_weights()
for ii in range(len(W)//2):
    print("Layer %d" %ii)
    print('Bias:\n', W[2*ii + 1])
    print('W:\n', W[2*ii])
    print()

plt.plot(history.history['loss'], label = "Train loss")
plt.plot(history.history['val_loss'], label = "Val loss")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()


plt.plot(history.history['accuracy'], label = "Train accuarcy")
plt.plot(history.history['val_accuracy'], label = "Val accuarcy")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()
```

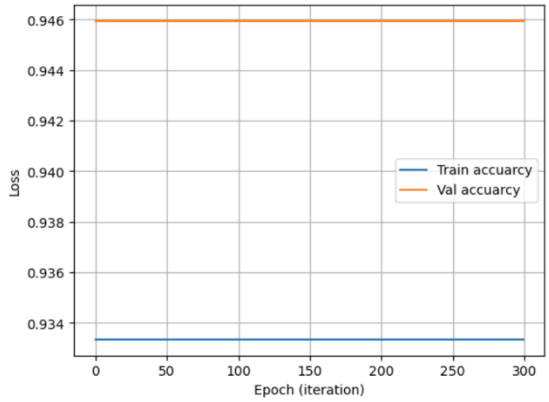
```
Test loss (cross-entropy and accuracy): [0.31758633255958557, 0.8947368264198303]
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)





4. model.compile(optimizer='adam',loss='BinaryCrossentropy.', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dump

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[62]:
```
model.compile(optimizer='Adam',loss='BinaryCrossentropy', metrics=['accuracy']) # compiling the model
history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9399 - loss: 0.1715 - val_accuracy: 0.9730 - val_loss: 0.1293
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9347 - loss: 0.1732 - val_accuracy: 0.9730 - val_loss: 0.1290
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9295 - loss: 0.1713 - val_accuracy: 0.9459 - val_loss: 0.1292
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1695 - val_accuracy: 0.9459 - val_loss: 0.1294
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1680 - val_accuracy: 0.9189 - val_loss: 0.1296
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1695 - val_accuracy: 0.9189 - val_loss: 0.1297
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1722 - val_accuracy: 0.9189 - val_loss: 0.1293
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1694 - val_accuracy: 0.9459 - val_loss: 0.1287
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9347 - loss: 0.1704 - val_accuracy: 0.9730 - val_loss: 0.1282
```

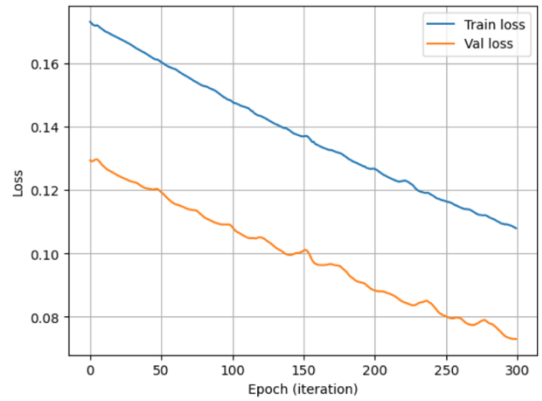
[63]:
```
loss = model.evaluate(X_test, Y_oh_test, verbose=0)
print('Test loss (cross-entropy and accuracy):',loss)
print()
W = model.get_weights()
for ii in range(len(W)//2):
    print("Layer %d" %ii)
    print('Bias:\n', W[2*ii + 1])
    print('W:\n', W[2*ii])
    print()

plt.plot(history.history['loss'], label = "Train loss")
plt.plot(history.history['val_loss'], label = "Val loss")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()


plt.plot(history.history['accuracy'], label = "Train accuarcy")
plt.plot(history.history['val_accuracy'], label = "Val accuarcy")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()
```

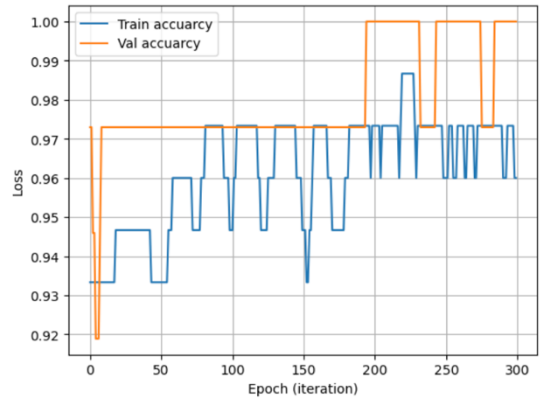
```
Test loss (cross-entropy and accuracy): [0.14857332408428192, 0.8947368264198303]
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)





5. model.compile(optimizer='adam',loss='CategoricalFocalCrossentropy', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dumps

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[66]:
```
model.compile(optimizer='adam',loss='CategoricalFocalCrossentropy', metrics=['accuracy']) # compiling the model
history = model.fit(X_train, Y_oh_train, validation_data=(X_val,Y_oh_val),batch_size= 64, epochs= 300)
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9577 - loss: 0.0033 - val_accuracy: 1.0000 - val_loss: 5.1643e-04
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9859 - loss: 0.0033 - val_accuracy: 1.0000 - val_loss: 5.5783e-04
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9859 - loss: 0.0033 - val_accuracy: 1.0000 - val_loss: 6.3587e-04
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9718 - loss: 0.0032 - val_accuracy: 1.0000 - val_loss: 6.5034e-04
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0032 - val_accuracy: 1.0000 - val_loss: 5.9653e-04
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9718 - loss: 0.0031 - val_accuracy: 1.0000 - val_loss: 5.9895e-04
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9770 - loss: 0.0029 - val_accuracy: 1.0000 - val_loss: 6.9284e-04
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0031 - val_accuracy: 1.0000 - val_loss: 8.1524e-04
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0033 - val_accuracy: 1.0000 - val_loss: 9.0550e-04
```

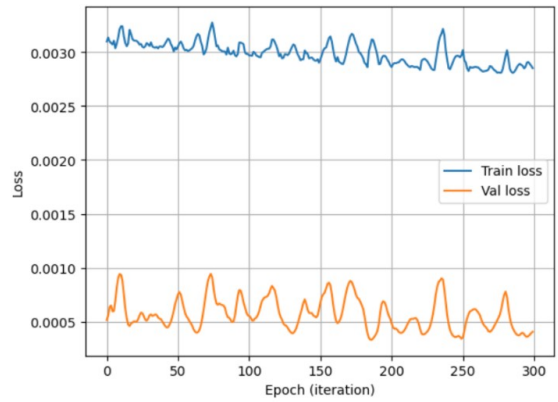
[67]:
```
loss = model.evaluate(X_test, Y_oh_test, verbose=0)
print('Test loss (cross-entropy and accuracy):',loss)
print()
W = model.get_weights()
for ii in range(len(W)//2):
    print("Layer %d" %ii)
    print('Bias:\n', W[2*ii + 1])
    print('W:\n', W[2*ii])
    print()

plt.plot(history.history['loss'], label = "Train loss")
plt.plot(history.history['val_loss'], label = "Val loss")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()


plt.plot(history.history['accuracy'], label = "Train accuarcy")
plt.plot(history.history['val_accuracy'], label = "Val accuarcy")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()
```

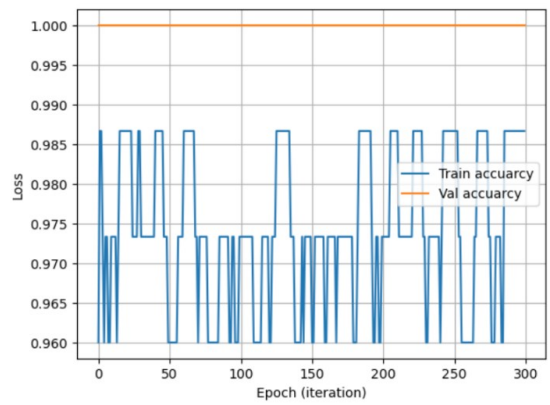
```
Test loss (cross-entropy and accuracy): [0.012138839811086655, 0.8947368264198303]
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 19 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)





6. model.compile(optimizer='adam',loss='SparseCategoricalCrossentropy', metrics=['accuracy']) # compiling the model

   Train model and execute loss = model.evaluate(X_test, Y_oh_test, verbose=0) print screen dumps

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[70]:
```
model.compile(optimizer='adam',loss='SparseCategoricalCrossentropy', metrics=['accuracy']) # compiling the model
history = model.fit(X_train, Y_train, validation_data=(X_val,Y_val),batch_size= 64, epochs= 300)
```

```
Epoch 21/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9859 - loss: 0.0770 - val_accuracy: 1.0000 - val_loss: 0.0338
Epoch 22/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9859 - loss: 0.0744 - val_accuracy: 1.0000 - val_loss: 0.0346
Epoch 23/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9770 - loss: 0.0736 - val_accuracy: 1.0000 - val_loss: 0.0354
Epoch 24/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.9718 - loss: 0.0757 - val_accuracy: 1.0000 - val_loss: 0.0356
Epoch 25/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9629 - loss: 0.0725 - val_accuracy: 1.0000 - val_loss: 0.0360
Epoch 26/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0806 - val_accuracy: 1.0000 - val_loss: 0.0362
Epoch 27/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0787 - val_accuracy: 1.0000 - val_loss: 0.0362
Epoch 28/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0796 - val_accuracy: 1.0000 - val_loss: 0.0360
Epoch 29/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9577 - loss: 0.0746 - val_accuracy: 1.0000 - val_loss: 0.0359
Epoch 30/300
```

[71]:
```
loss = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss (cross-entropy and accuracy):',loss)
print()
W = model.get_weights()
for ii in range(len(W)//2):
    print("Layer %d" %ii)
    print('Bias:\n', W[2*ii + 1])
    print('W:\n', W[2*ii])
    print()

plt.plot(history.history['loss'], label = "Train loss")
plt.plot(history.history['val_loss'], label = "Val loss")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()


plt.plot(history.history['accuracy'], label = "Train accuarcy")
plt.plot(history.history['val_accuracy'], label = "Val accuarcy")
plt.xlabel("Epoch (iteration)")
plt.ylabel("Loss")
plt.legend()
plt.grid()
plt.show()
```

```
Test loss (cross-entropy and accuracy): [0.13839274644851685, 0.9210526347160339]
```

jupyter Modify Parameters_ Shruti Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)