DenkowskiStanislawlab6

December 2, 2022

[ ]:
```
!pip install pycocotools --user
```

```
Collecting pycocotools
  Downloading pycocotools-2.0.6.tar.gz (24 kB)
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Preparing metadata (pyproject.toml) … done
Requirement already satisfied: matplotlib>=2.1.0 in
/opt/conda/lib/python3.7/site-packages (from pycocotools) (3.5.3)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from pycocotools) (1.21.6)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib>=2.1.0->pycocotools) (0.11.0)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.7/site-
packages (from matplotlib>=2.1.0->pycocotools) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.0->pycocotools)
(3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.0->pycocotools)
(4.33.3)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.7/site-
packages (from matplotlib>=2.1.0->pycocotools) (9.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.0->pycocotools)
(2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.0->pycocotools)
(1.4.3)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.7/site-packages (from
kiwisolver>=1.0.1->matplotlib>=2.1.0->pycocotools) (4.1.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-
packages (from python-dateutil>=2.7->matplotlib>=2.1.0->pycocotools) (1.15.0)
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (pyproject.toml) … done
  Created wheel for pycocotools:
filename=pycocotools-2.0.6-cp37-cp37m-linux_x86_64.whl size=373761
```

```
sha256=5d9fa500d156cb59ea01a5c29b8c7b58963c3aa8d7d96dd0c0bec0e3e7d5f9d9
  Stored in directory: /root/.cache/pip/wheels/06/f6/f9/9cc49c6de8e3cf27dfddd91b
f46595a057141d4583a2adaf03
Successfully built pycocotools
Installing collected packages: pycocotools
Successfully installed pycocotools-2.0.6
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

[ ]:
```python
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import tensorflow as tf
import tensorflow_addons as tfa
import numpy as np
import os
import time
```

[ ]:
```python
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.
 ↪cifar10.load_data()
classes = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',␣
 ↪'horse', 'ship', 'truck']
train_labels, test_labels = tf.squeeze(tf.one_hot(train_labels, len(classes))),␣
 ↪tf.squeeze(tf.one_hot(test_labels, len(classes)))
train_images, test_images = tf.convert_to_tensor(train_images), tf.
 ↪convert_to_tensor(test_images)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 4s 0us/step
170508288/170498071 [==============================] - 4s 0us/step
```

[ ]:
```python
augment = tf.keras.Sequential([
    tf.keras.layers.RandomFlip(mode='horizontal'),
    tf.keras.layers.RandomZoom((-0.1,0.1)),
    tf.keras.layers.Normalization()
])
augment.layers[-1].adapt(train_images)
```

```
2022-12-02 14:22:48.567276: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR
Optimization Passes are enabled (registered 2)
```

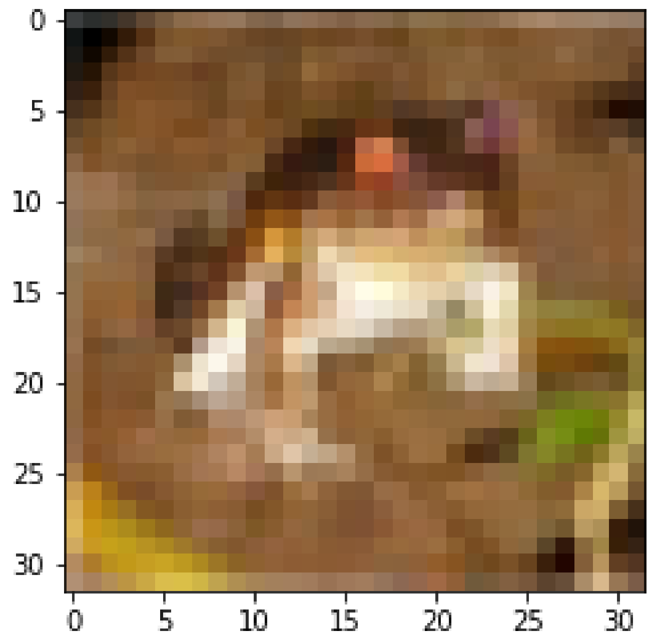
Wczytałem Cifar10, następnie przygotowałem model, który wykonuje augmentację w postaci RandomFlip, RandomZoom oraz normalizację, ponieważ niestety tensorflow nie ma RandomResizedCrop a rozwiązania jakie znalazłem nie działały.
W tensorflow, z tego co rozumiem to albo wrzuca się przez to dane, albo podpina na początek

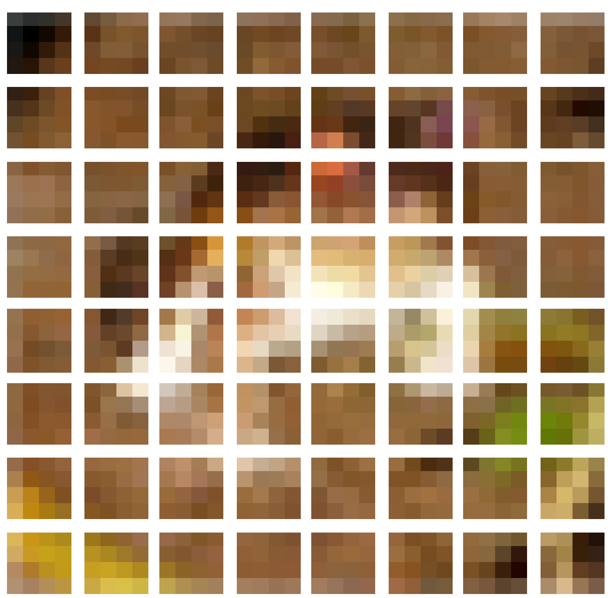
modelu i tak spróbuję(choć ostatnio miałem problem).
Dodatkowo zaaplikowałem one hot encoding na labele i zrzutowałem obrazki na tensor.

[ ]:
```
class Patches(tf.keras.layers.Layer):
    def __init__(self, size):
        super().__init__()
        self.size = size

    def call(self, images):
        batch_size = tf.shape(images)[0]
        patches = tf.image.extract_patches(
            images=images,
            sizes=[1,self.size,self.size,1],
            strides=[1,self.size,self.size,1],
            rates=[1,1,1,1],
            padding='VALID'
        )
        patch_dims = patches.shape[-1]
        patches = tf.reshape(patches, [batch_size, -1, patch_dims])
        return patches
```

Klasa Patches zawiera implementację rozkładania obrazka na patche, wykorzystuję do tego tf.image.extract_patches.
Warto zauważyć, że klasa dziedziczy z tf.keras.layers.Layer, dzięki czemu może być częścią modelu.

[ ]:
```
patches = Patches(4)
plt.imshow(train_images[0])
plt.show()
patched = patches(tf.expand_dims(train_images[0],0))

plt.figure(figsize=(4,4))
for ix, im in enumerate(patched[0]):
    plt.subplot(8, 8, ix + 1)
    im = tf.reshape(im, (4,4,3))
    plt.imshow(im)
    plt.axis('off')
```





[ ]:
```
class Encoder(tf.keras.layers.Layer):
    def __init__(self, num_patches=8*8, projection_dim=256):
        super().__init__()
        self.num_patches = num_patches
        self.projection_dim = projection_dim
```

```
        init = tf.random_normal_initializer()
        class_token = init(shape=(1,self.projection_dim), dtype=tf.float32)
        self.class_token = tf.Variable(initial_value=class_token,␣
↪trainable=True)
        self.projection = tf.keras.layers.Dense(units=self.projection_dim)
        self.positional_embedding = tf.keras.layers.Embedding(input_dim=self.
↪num_patches+1, output_dim=self.projection_dim)

    def call(self, batch):
        batch_size = tf.shape(batch)[0]
        class_token = tf.tile(input=self.class_token, multiples=[batch_size,1])
        class_token = tf.reshape(class_token, shape=(batch_size,1,self.
↪projection_dim))
        embs = self.projection(batch)
        embs = tf.concat([class_token, embs], 1)
        positions = tf.range(0, self.num_patches+1, 1)
        pos_embs = self.positional_embedding(positions)
        encoded = embs+pos_embs
        return encoded
```

Do stworzenia encodera użyłem: - tf.random_normal_initializer - do zainicjowania class tokena, inicjuję go trochę jako batch, dlatego wymiar (1,rozmiar tokena) - tf.Variable - aby class token się uczył i był używany przez framework - tf.keras.layers.Dense - do liniowego embeddingu patchy - tf.keras.layers.Embedding - do embeddingu pozycyjnego

Ponadto do wykorzystania go używam: - tf.tile - aby dodać class_token do każdego obrazka w batchu - tf.reshape - aby batch class_tokenów miał właściwe wymiary - warstwy dense, wcześniej zainicjowanej - do liniowego embeddingu patchy - tf.concat - aby dokleić do batcha class_tokeny - tf.range - aby wyznaczyć embeddingi pozycyjne - warstwy embedding, wcześniej zainicjowanej - do wyznaczenia embeddingu pozycyjnego - ona się będzie uczyć, ale potrzebuje dostać tf.range - sumuję embeddingi i otrzymuję encoding

[ ]:
```
class Transformer(tf.keras.layers.Layer):
    def __init__(self, num_head=8, projection_dim=256, hidden_units=512):
        super().__init__()
        self.num_head = num_head
        self.projection_dim = projection_dim
        self.hidden_units = hidden_units
        self.norm1 = tf.keras.layers.LayerNormalization()
        self.multihead = tf.keras.layers.MultiHeadAttention(num_heads=self.
↪num_head, key_dim=self.projection_dim)
        self.norm2 = tf.keras.layers.LayerNormalization()
        self.linear1 = tf.keras.layers.Dense(units=self.hidden_units,␣
↪activation=tf.nn.gelu)
        self.dropout1 = tf.keras.layers.Dropout(rate=0.2)
        self.linear2 = tf.keras.layers.Dense(units=self.projection_dim)
        self.dropout2 = tf.keras.layers.Dropout(rate=0.2)
```

```python
    def call(self, batch):
        batch1 = self.norm1(batch)
        attentioned = self.multihead(batch1, batch1)
        self.attention_output = attentioned
        batch2 = tf.keras.layers.Add()([attentioned, batch])
        batch3 = self.norm2(batch2)
        batch3 = self.linear1(batch3)
        batch3 = self.dropout1(batch3)
        batch3 = self.linear2(batch3)
        batch3 = self.dropout2(batch3)
        batch4 = tf.keras.layers.Add()([batch3, batch2])
        return batch4
```

Aby stworzy blok transformera użyłem: - tf.keras.layers.LayerNormalization - jako warstwy normalizującej - tf.keras.layers.MultiHeadAttention - jako warstwy multi head attention - tf.keras.layers.Dense - Jako liniowej warstwy - tf.keras.layers.Dropout - Jako warstwy dropout - tf.nn.gelu - jako funkcji aktywacji

Do wykorzystania użyłem jeszcze: - tf.keras.layers.Add - jako połączenie rezydualne

[ ]:
```python
def create_vit(patch_size=4, transformers_num=6):
    inputs = tf.keras.layers.Input(shape=(32,32,3))
    # augmented = augment(inputs)
    patches = Patches(size=patch_size)(inputs)#(augmented)
    embedded = Encoder()(patches)
    transformed = tf.keras.layers.Dropout(0.2)(embedded)
    for _ in range(transformers_num):
        transformed = Transformer()(transformed)
    normed = tf.keras.layers.LayerNormalization()(transformed[:,0])
    mlped = tf.keras.layers.Dense(units=len(classes),␣
 ↪activation='softmax')(normed)
    model = tf.keras.Model(inputs=inputs, outputs=mlped)
    return model

vit = create_vit()
```

Aby zdefiniować model, ustalam input, przepuszczam input przez augmentację, następnie rozkładam obrazki na patche, embedduje, nakładam bloki transformerów 6 razy, a na koniec jedna warstwa liniowa.

[ ]:
```python
vit.summary()
```

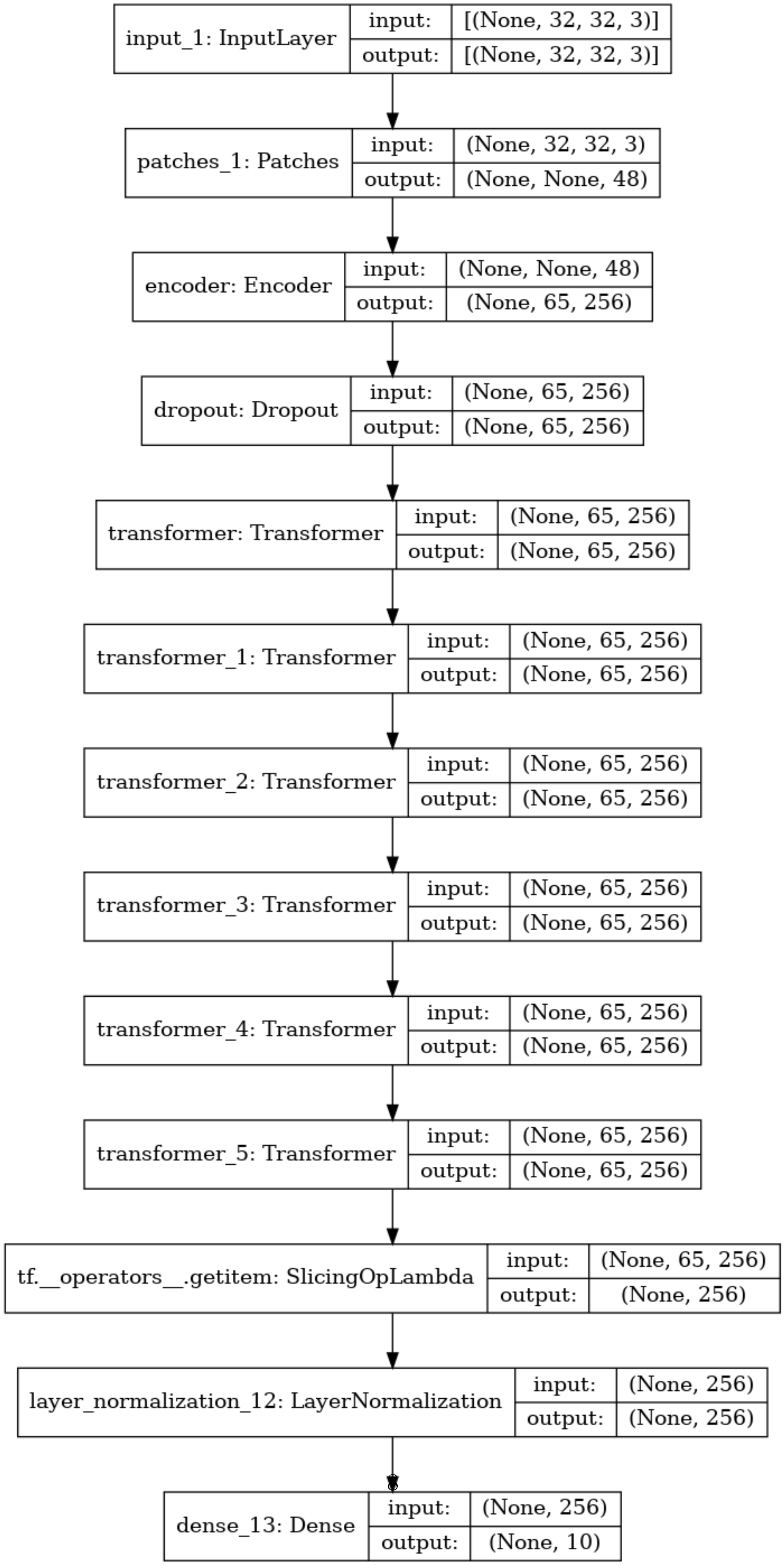
```
Model: "model"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 32, 32, 3)]       0

_________________________________________________________________
patches_1 (Patches)          (None, None, 48)          0
```

```
_________________________________________________________________
encoder (Encoder)            (None, 65, 256)           29440
_________________________________________________________________
dropout (Dropout)            (None, 65, 256)           0
_________________________________________________________________
transformer (Transformer)    (None, 65, 256)           2367488
_________________________________________________________________
transformer_1 (Transformer)  (None, 65, 256)           2367488
_________________________________________________________________
transformer_2 (Transformer)  (None, 65, 256)           2367488
_________________________________________________________________
transformer_3 (Transformer)  (None, 65, 256)           2367488
_________________________________________________________________
transformer_4 (Transformer)  (None, 65, 256)           2367488
_________________________________________________________________
transformer_5 (Transformer)  (None, 65, 256)           2367488
_________________________________________________________________
tf.__operators__.getitem (Sl (None, 256)               0
_________________________________________________________________
layer_normalization_12 (Laye (None, 256)               512
_________________________________________________________________
dense_13 (Dense)             (None, 10)                2570
=================================================================
Total params: 14,237,450
Trainable params: 14,237,450
Non-trainable params: 0
_________________________________________________________________
```

[ ]:

```python
tf.keras.utils.plot_model(vit, to_file='./transformer.png', show_shapes=True,
    expand_nested=True)
```

[ ]:


input_1: InputLayer
input: [(None, 32, 32, 3)]
output: [(None, 32, 32, 3)]
patches_1: Patches
input: (None, 32, 32, 3)
output: (None, None, 48)
encoder: Encoder
input: (None, None, 48)
output: (None, 65, 256)
dropout: Dropout
input: (None, 65, 256)
output: (None, 65, 256)
transformer: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
transformer_1: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
transformer_2: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
transformer_3: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
transformer_4: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
transformer_5: Transformer
input: (None, 65, 256)
output: (None, 65, 256)
tf.__operators__.getitem: SlicingOpLambda
input: (None, 65, 256)
output: (None, 256)
layer_normalization_12: LayerNormalization
input: (None, 256)
output: (None, 256)
dense_13: Dense
input: (None, 256)
output: (None, 10)

Niestety nie umiem rozwinąć wizualizacji bloków transformer i encoder, ale wydaje mi się, że powinny być dobrze.

[ ]:
```
def scheduler(epoch, lr):
    if epoch == 100 or epoch == 150:
        return lr * tf.constant(0.1)
    else:
        return lr

cb = tf.keras.callbacks.LearningRateScheduler(scheduler)
```

[ ]:
```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.00001)
vit.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.CategoricalCrossentropy(),
    metrics=['accuracy']
)
round(vit.optimizer.lr.numpy(), 5)
```

[ ]:
```
1e-05
```

[ ]:
```
history = vit.fit(
    x=train_images,
    y=train_labels,
    batch_size=128,
    epochs=160,
    callbacks=[cb],
    validation_data=(test_images, test_labels)
)
round(vit.optimizer.lr.numpy(), 5)
```

```
Epoch 1/160
```

```
2022-12-02 14:23:03.208208: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369]
Loaded cuDNN version 8005
```

```
391/391 [==============================] - 109s 260ms/step - loss: 2.3080 -
accuracy: 0.1619 - val_loss: 2.0736 - val_accuracy: 0.2101
Epoch 2/160
391/391 [==============================] - 101s 258ms/step - loss: 2.0804 -
accuracy: 0.2134 - val_loss: 1.9628 - val_accuracy: 0.2706
Epoch 3/160
391/391 [==============================] - 101s 258ms/step - loss: 1.9682 -
accuracy: 0.2552 - val_loss: 1.8661 - val_accuracy: 0.2949
Epoch 4/160
391/391 [==============================] - 101s 258ms/step - loss: 1.8961 -
accuracy: 0.2779 - val_loss: 1.8080 - val_accuracy: 0.3244
```

```
Epoch 5/160
391/391 [==============================] - 101s 258ms/step - loss: 1.8415 -
accuracy: 0.3012 - val_loss: 1.7726 - val_accuracy: 0.3459
Epoch 6/160
391/391 [==============================] - 101s 257ms/step - loss: 1.7885 -
accuracy: 0.3231 - val_loss: 1.7151 - val_accuracy: 0.3696
Epoch 7/160
391/391 [==============================] - 101s 258ms/step - loss: 1.7409 -
accuracy: 0.3450 - val_loss: 1.6820 - val_accuracy: 0.3862
Epoch 8/160
391/391 [==============================] - 101s 258ms/step - loss: 1.6965 -
accuracy: 0.3665 - val_loss: 1.6370 - val_accuracy: 0.4024
Epoch 9/160
391/391 [==============================] - 101s 258ms/step - loss: 1.6539 -
accuracy: 0.3876 - val_loss: 1.5997 - val_accuracy: 0.4186
Epoch 10/160
391/391 [==============================] - 101s 258ms/step - loss: 1.6101 -
accuracy: 0.4106 - val_loss: 1.5515 - val_accuracy: 0.4326
Epoch 11/160
391/391 [==============================] - 101s 258ms/step - loss: 1.5679 -
accuracy: 0.4293 - val_loss: 1.5184 - val_accuracy: 0.4463
Epoch 12/160
391/391 [==============================] - 101s 258ms/step - loss: 1.5330 -
accuracy: 0.4423 - val_loss: 1.4945 - val_accuracy: 0.4589
Epoch 13/160
391/391 [==============================] - 101s 258ms/step - loss: 1.4989 -
accuracy: 0.4576 - val_loss: 1.4604 - val_accuracy: 0.4738
Epoch 14/160
391/391 [==============================] - 101s 258ms/step - loss: 1.4632 -
accuracy: 0.4727 - val_loss: 1.4343 - val_accuracy: 0.4763
Epoch 15/160
391/391 [==============================] - 101s 258ms/step - loss: 1.4377 -
accuracy: 0.4827 - val_loss: 1.4514 - val_accuracy: 0.4749
Epoch 16/160
391/391 [==============================] - 101s 258ms/step - loss: 1.4132 -
accuracy: 0.4932 - val_loss: 1.4096 - val_accuracy: 0.4925
Epoch 17/160
391/391 [==============================] - 101s 257ms/step - loss: 1.3875 -
accuracy: 0.5013 - val_loss: 1.3966 - val_accuracy: 0.4988
Epoch 18/160
391/391 [==============================] - 101s 257ms/step - loss: 1.3703 -
accuracy: 0.5084 - val_loss: 1.3685 - val_accuracy: 0.5089
Epoch 19/160
391/391 [==============================] - 100s 256ms/step - loss: 1.3471 -
accuracy: 0.5186 - val_loss: 1.3636 - val_accuracy: 0.5136
Epoch 20/160
391/391 [==============================] - 100s 257ms/step - loss: 1.3327 -
accuracy: 0.5254 - val_loss: 1.3563 - val_accuracy: 0.5131
```

```
Epoch 21/160
391/391 [==============================] - 100s 257ms/step - loss: 1.3194 -
accuracy: 0.5279 - val_loss: 1.3415 - val_accuracy: 0.5179
Epoch 22/160
391/391 [==============================] - 100s 256ms/step - loss: 1.3052 -
accuracy: 0.5345 - val_loss: 1.3319 - val_accuracy: 0.5213
Epoch 23/160
391/391 [==============================] - 100s 256ms/step - loss: 1.2883 -
accuracy: 0.5385 - val_loss: 1.3307 - val_accuracy: 0.5210
Epoch 24/160
391/391 [==============================] - 101s 257ms/step - loss: 1.2754 -
accuracy: 0.5436 - val_loss: 1.3264 - val_accuracy: 0.5253
Epoch 25/160
391/391 [==============================] - 101s 258ms/step - loss: 1.2613 -
accuracy: 0.5511 - val_loss: 1.3046 - val_accuracy: 0.5306
Epoch 26/160
391/391 [==============================] - 100s 257ms/step - loss: 1.2551 -
accuracy: 0.5509 - val_loss: 1.2874 - val_accuracy: 0.5386
Epoch 27/160
391/391 [==============================] - 100s 257ms/step - loss: 1.2427 -
accuracy: 0.5555 - val_loss: 1.2958 - val_accuracy: 0.5348
Epoch 28/160
391/391 [==============================] - 100s 257ms/step - loss: 1.2283 -
accuracy: 0.5621 - val_loss: 1.2838 - val_accuracy: 0.5392
Epoch 29/160
391/391 [==============================] - 100s 257ms/step - loss: 1.2181 -
accuracy: 0.5645 - val_loss: 1.2856 - val_accuracy: 0.5370
Epoch 30/160
391/391 [==============================] - 100s 256ms/step - loss: 1.2099 -
accuracy: 0.5694 - val_loss: 1.2893 - val_accuracy: 0.5380
Epoch 31/160
391/391 [==============================] - 100s 256ms/step - loss: 1.2011 -
accuracy: 0.5735 - val_loss: 1.2776 - val_accuracy: 0.5431
Epoch 32/160
391/391 [==============================] - 100s 256ms/step - loss: 1.1906 -
accuracy: 0.5745 - val_loss: 1.2643 - val_accuracy: 0.5477
Epoch 33/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1816 -
accuracy: 0.5784 - val_loss: 1.2536 - val_accuracy: 0.5506
Epoch 34/160
391/391 [==============================] - 101s 258ms/step - loss: 1.1715 -
accuracy: 0.5828 - val_loss: 1.2453 - val_accuracy: 0.5557
Epoch 35/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1632 -
accuracy: 0.5852 - val_loss: 1.2368 - val_accuracy: 0.5591
Epoch 36/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1561 -
accuracy: 0.5889 - val_loss: 1.2267 - val_accuracy: 0.5607
```

```
Epoch 37/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1501 -
accuracy: 0.5926 - val_loss: 1.2304 - val_accuracy: 0.5595
Epoch 38/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1429 -
accuracy: 0.5925 - val_loss: 1.2300 - val_accuracy: 0.5639
Epoch 39/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1343 -
accuracy: 0.5959 - val_loss: 1.2207 - val_accuracy: 0.5672
Epoch 40/160
391/391 [==============================] - 100s 256ms/step - loss: 1.1249 -
accuracy: 0.5991 - val_loss: 1.2274 - val_accuracy: 0.5630
Epoch 41/160
391/391 [==============================] - 100s 256ms/step - loss: 1.1199 -
accuracy: 0.6009 - val_loss: 1.2143 - val_accuracy: 0.5678
Epoch 42/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1122 -
accuracy: 0.6032 - val_loss: 1.2117 - val_accuracy: 0.5681
Epoch 43/160
391/391 [==============================] - 100s 257ms/step - loss: 1.1069 -
accuracy: 0.6059 - val_loss: 1.2073 - val_accuracy: 0.5708
Epoch 44/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0983 -
accuracy: 0.6069 - val_loss: 1.2176 - val_accuracy: 0.5673
Epoch 45/160
391/391 [==============================] - 100s 256ms/step - loss: 1.0917 -
accuracy: 0.6119 - val_loss: 1.2094 - val_accuracy: 0.5726
Epoch 46/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0882 -
accuracy: 0.6118 - val_loss: 1.1961 - val_accuracy: 0.5774
Epoch 47/160
391/391 [==============================] - 101s 257ms/step - loss: 1.0776 -
accuracy: 0.6156 - val_loss: 1.2013 - val_accuracy: 0.5703
Epoch 48/160
391/391 [==============================] - 101s 258ms/step - loss: 1.0715 -
accuracy: 0.6187 - val_loss: 1.1955 - val_accuracy: 0.5762
Epoch 49/160
391/391 [==============================] - 101s 257ms/step - loss: 1.0631 -
accuracy: 0.6212 - val_loss: 1.1909 - val_accuracy: 0.5791
Epoch 50/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0612 -
accuracy: 0.6207 - val_loss: 1.1798 - val_accuracy: 0.5792
Epoch 51/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0558 -
accuracy: 0.6246 - val_loss: 1.1812 - val_accuracy: 0.5807
Epoch 52/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0474 -
accuracy: 0.6286 - val_loss: 1.1721 - val_accuracy: 0.5841
```

```
Epoch 53/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0458 -
accuracy: 0.6261 - val_loss: 1.1722 - val_accuracy: 0.5831
Epoch 54/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0376 -
accuracy: 0.6276 - val_loss: 1.1898 - val_accuracy: 0.5819
Epoch 55/160
391/391 [==============================] - 100s 256ms/step - loss: 1.0307 -
accuracy: 0.6306 - val_loss: 1.1815 - val_accuracy: 0.5818
Epoch 56/160
391/391 [==============================] - 100s 256ms/step - loss: 1.0281 -
accuracy: 0.6345 - val_loss: 1.1680 - val_accuracy: 0.5865
Epoch 57/160
391/391 [==============================] - 100s 256ms/step - loss: 1.0181 -
accuracy: 0.6374 - val_loss: 1.1998 - val_accuracy: 0.5791
Epoch 58/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0154 -
accuracy: 0.6376 - val_loss: 1.1809 - val_accuracy: 0.5837
Epoch 59/160
391/391 [==============================] - 100s 257ms/step - loss: 1.0100 -
accuracy: 0.6371 - val_loss: 1.1851 - val_accuracy: 0.5815
Epoch 60/160
391/391 [==============================] - 100s 256ms/step - loss: 1.0050 -
accuracy: 0.6410 - val_loss: 1.1712 - val_accuracy: 0.5859
Epoch 61/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9972 -
accuracy: 0.6423 - val_loss: 1.1681 - val_accuracy: 0.5874
Epoch 62/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9908 -
accuracy: 0.6459 - val_loss: 1.1701 - val_accuracy: 0.5885
Epoch 63/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9879 -
accuracy: 0.6497 - val_loss: 1.1735 - val_accuracy: 0.5852
Epoch 64/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9828 -
accuracy: 0.6490 - val_loss: 1.2074 - val_accuracy: 0.5796
Epoch 65/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9734 -
accuracy: 0.6524 - val_loss: 1.1737 - val_accuracy: 0.5878
Epoch 66/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9704 -
accuracy: 0.6524 - val_loss: 1.1649 - val_accuracy: 0.5910
Epoch 67/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9652 -
accuracy: 0.6561 - val_loss: 1.1656 - val_accuracy: 0.5913
Epoch 68/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9603 -
accuracy: 0.6555 - val_loss: 1.1602 - val_accuracy: 0.5925
```

```
Epoch 69/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9554 -
accuracy: 0.6603 - val_loss: 1.1559 - val_accuracy: 0.5939
Epoch 70/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9485 -
accuracy: 0.6601 - val_loss: 1.1754 - val_accuracy: 0.5895
Epoch 71/160
391/391 [==============================] - 100s 256ms/step - loss: 0.9465 -
accuracy: 0.6617 - val_loss: 1.1756 - val_accuracy: 0.5921
Epoch 72/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9411 -
accuracy: 0.6623 - val_loss: 1.1568 - val_accuracy: 0.5971
Epoch 73/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9356 -
accuracy: 0.6649 - val_loss: 1.1805 - val_accuracy: 0.5910
Epoch 74/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9310 -
accuracy: 0.6670 - val_loss: 1.1663 - val_accuracy: 0.5969
Epoch 75/160
391/391 [==============================] - 100s 256ms/step - loss: 0.9244 -
accuracy: 0.6720 - val_loss: 1.1661 - val_accuracy: 0.5952
Epoch 76/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9201 -
accuracy: 0.6708 - val_loss: 1.1677 - val_accuracy: 0.5976
Epoch 77/160
391/391 [==============================] - 100s 256ms/step - loss: 0.9140 -
accuracy: 0.6750 - val_loss: 1.1774 - val_accuracy: 0.5933
Epoch 78/160
391/391 [==============================] - 100s 256ms/step - loss: 0.9084 -
accuracy: 0.6742 - val_loss: 1.1619 - val_accuracy: 0.6008
Epoch 79/160
391/391 [==============================] - 100s 257ms/step - loss: 0.9026 -
accuracy: 0.6782 - val_loss: 1.1840 - val_accuracy: 0.5912
Epoch 80/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8929 -
accuracy: 0.6803 - val_loss: 1.1845 - val_accuracy: 0.5941
Epoch 81/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8898 -
accuracy: 0.6827 - val_loss: 1.1735 - val_accuracy: 0.5979
Epoch 82/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8902 -
accuracy: 0.6814 - val_loss: 1.1622 - val_accuracy: 0.5988
Epoch 83/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8888 -
accuracy: 0.6820 - val_loss: 1.1737 - val_accuracy: 0.5998
Epoch 84/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8786 -
accuracy: 0.6861 - val_loss: 1.1657 - val_accuracy: 0.5963
```

```
Epoch 85/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8721 -
accuracy: 0.6861 - val_loss: 1.1778 - val_accuracy: 0.5995
Epoch 86/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8690 -
accuracy: 0.6902 - val_loss: 1.1719 - val_accuracy: 0.6025
Epoch 87/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8620 -
accuracy: 0.6903 - val_loss: 1.1737 - val_accuracy: 0.5991
Epoch 88/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8552 -
accuracy: 0.6940 - val_loss: 1.1703 - val_accuracy: 0.5987
Epoch 89/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8536 -
accuracy: 0.6939 - val_loss: 1.1666 - val_accuracy: 0.6026
Epoch 90/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8469 -
accuracy: 0.6977 - val_loss: 1.1771 - val_accuracy: 0.5966
Epoch 91/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8399 -
accuracy: 0.6984 - val_loss: 1.1792 - val_accuracy: 0.6020
Epoch 92/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8335 -
accuracy: 0.7041 - val_loss: 1.1722 - val_accuracy: 0.6012
Epoch 93/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8327 -
accuracy: 0.7010 - val_loss: 1.1831 - val_accuracy: 0.5999
Epoch 94/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8247 -
accuracy: 0.7027 - val_loss: 1.1893 - val_accuracy: 0.5979
Epoch 95/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8221 -
accuracy: 0.7060 - val_loss: 1.1883 - val_accuracy: 0.5965
Epoch 96/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8172 -
accuracy: 0.7067 - val_loss: 1.1991 - val_accuracy: 0.5977
Epoch 97/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8096 -
accuracy: 0.7116 - val_loss: 1.1856 - val_accuracy: 0.6005
Epoch 98/160
391/391 [==============================] - 100s 256ms/step - loss: 0.8077 -
accuracy: 0.7106 - val_loss: 1.1916 - val_accuracy: 0.6011
Epoch 99/160
391/391 [==============================] - 100s 257ms/step - loss: 0.8026 -
accuracy: 0.7116 - val_loss: 1.1918 - val_accuracy: 0.6032
Epoch 100/160
391/391 [==============================] - 104s 265ms/step - loss: 0.7951 -
accuracy: 0.7149 - val_loss: 1.1936 - val_accuracy: 0.6001
```

```
Epoch 101/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7674 -
accuracy: 0.7263 - val_loss: 1.1883 - val_accuracy: 0.6035
Epoch 102/160
391/391 [==============================] - 100s 256ms/step - loss: 0.7637 -
accuracy: 0.7280 - val_loss: 1.1903 - val_accuracy: 0.6023
Epoch 103/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7609 -
accuracy: 0.7281 - val_loss: 1.1911 - val_accuracy: 0.6063
Epoch 104/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7621 -
accuracy: 0.7284 - val_loss: 1.1924 - val_accuracy: 0.6058
Epoch 105/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7597 -
accuracy: 0.7284 - val_loss: 1.1985 - val_accuracy: 0.6045
Epoch 106/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7564 -
accuracy: 0.7305 - val_loss: 1.1914 - val_accuracy: 0.6060
Epoch 107/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7582 -
accuracy: 0.7282 - val_loss: 1.1929 - val_accuracy: 0.6053
Epoch 108/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7565 -
accuracy: 0.7292 - val_loss: 1.1963 - val_accuracy: 0.6051
Epoch 109/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7550 -
accuracy: 0.7303 - val_loss: 1.1955 - val_accuracy: 0.6047
Epoch 110/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7559 -
accuracy: 0.7276 - val_loss: 1.1996 - val_accuracy: 0.6056
Epoch 111/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7545 -
accuracy: 0.7291 - val_loss: 1.1966 - val_accuracy: 0.6033
Epoch 112/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7550 -
accuracy: 0.7278 - val_loss: 1.1923 - val_accuracy: 0.6040
Epoch 113/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7527 -
accuracy: 0.7301 - val_loss: 1.1979 - val_accuracy: 0.6054
Epoch 114/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7508 -
accuracy: 0.7314 - val_loss: 1.1995 - val_accuracy: 0.6052
Epoch 115/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7498 -
accuracy: 0.7311 - val_loss: 1.2039 - val_accuracy: 0.6045
Epoch 116/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7551 -
accuracy: 0.7278 - val_loss: 1.1973 - val_accuracy: 0.6033
```

```
Epoch 117/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7526 -
accuracy: 0.7310 - val_loss: 1.2064 - val_accuracy: 0.6051
Epoch 118/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7487 -
accuracy: 0.7317 - val_loss: 1.1983 - val_accuracy: 0.6049
Epoch 119/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7492 -
accuracy: 0.7314 - val_loss: 1.1990 - val_accuracy: 0.6064
Epoch 120/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7489 -
accuracy: 0.7310 - val_loss: 1.1975 - val_accuracy: 0.6051
Epoch 121/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7455 -
accuracy: 0.7332 - val_loss: 1.2065 - val_accuracy: 0.6044
Epoch 122/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7466 -
accuracy: 0.7329 - val_loss: 1.1993 - val_accuracy: 0.6062
Epoch 123/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7492 -
accuracy: 0.7317 - val_loss: 1.2004 - val_accuracy: 0.6060
Epoch 124/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7469 -
accuracy: 0.7326 - val_loss: 1.2022 - val_accuracy: 0.6052
Epoch 125/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7469 -
accuracy: 0.7339 - val_loss: 1.2035 - val_accuracy: 0.6033
Epoch 126/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7452 -
accuracy: 0.7337 - val_loss: 1.2045 - val_accuracy: 0.6045
Epoch 127/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7448 -
accuracy: 0.7327 - val_loss: 1.2033 - val_accuracy: 0.6039
Epoch 128/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7442 -
accuracy: 0.7323 - val_loss: 1.2096 - val_accuracy: 0.6040
Epoch 129/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7407 -
accuracy: 0.7337 - val_loss: 1.2100 - val_accuracy: 0.6047
Epoch 130/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7426 -
accuracy: 0.7329 - val_loss: 1.2070 - val_accuracy: 0.6023
Epoch 131/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7443 -
accuracy: 0.7328 - val_loss: 1.2046 - val_accuracy: 0.6042
Epoch 132/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7358 -
accuracy: 0.7368 - val_loss: 1.2073 - val_accuracy: 0.6012
```

```
Epoch 133/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7401 -
accuracy: 0.7334 - val_loss: 1.2089 - val_accuracy: 0.6045
Epoch 134/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7418 -
accuracy: 0.7350 - val_loss: 1.2073 - val_accuracy: 0.6033
Epoch 135/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7384 -
accuracy: 0.7361 - val_loss: 1.2106 - val_accuracy: 0.6043
Epoch 136/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7372 -
accuracy: 0.7347 - val_loss: 1.2083 - val_accuracy: 0.6042
Epoch 137/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7396 -
accuracy: 0.7353 - val_loss: 1.2124 - val_accuracy: 0.6042
Epoch 138/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7389 -
accuracy: 0.7359 - val_loss: 1.2079 - val_accuracy: 0.6044
Epoch 139/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7384 -
accuracy: 0.7355 - val_loss: 1.2044 - val_accuracy: 0.6037
Epoch 140/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7370 -
accuracy: 0.7354 - val_loss: 1.2070 - val_accuracy: 0.6056
Epoch 141/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7363 -
accuracy: 0.7351 - val_loss: 1.2114 - val_accuracy: 0.6061
Epoch 142/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7329 -
accuracy: 0.7395 - val_loss: 1.2129 - val_accuracy: 0.6044
Epoch 143/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7344 -
accuracy: 0.7363 - val_loss: 1.2079 - val_accuracy: 0.6032
Epoch 144/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7362 -
accuracy: 0.7391 - val_loss: 1.2150 - val_accuracy: 0.6038
Epoch 145/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7339 -
accuracy: 0.7377 - val_loss: 1.2094 - val_accuracy: 0.6051
Epoch 146/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7331 -
accuracy: 0.7354 - val_loss: 1.2124 - val_accuracy: 0.6043
Epoch 147/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7328 -
accuracy: 0.7367 - val_loss: 1.2087 - val_accuracy: 0.6054
Epoch 148/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7362 -
accuracy: 0.7354 - val_loss: 1.2148 - val_accuracy: 0.6018
```

```
Epoch 149/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7314 -
accuracy: 0.7370 - val_loss: 1.2113 - val_accuracy: 0.6051
Epoch 150/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7326 -
accuracy: 0.7363 - val_loss: 1.2142 - val_accuracy: 0.6047
Epoch 151/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7292 -
accuracy: 0.7394 - val_loss: 1.2122 - val_accuracy: 0.6044
Epoch 152/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7269 -
accuracy: 0.7409 - val_loss: 1.2112 - val_accuracy: 0.6046
Epoch 153/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7274 -
accuracy: 0.7382 - val_loss: 1.2113 - val_accuracy: 0.6053
Epoch 154/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7274 -
accuracy: 0.7403 - val_loss: 1.2118 - val_accuracy: 0.6048
Epoch 155/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7257 -
accuracy: 0.7390 - val_loss: 1.2122 - val_accuracy: 0.6045
Epoch 156/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7245 -
accuracy: 0.7397 - val_loss: 1.2124 - val_accuracy: 0.6055
Epoch 157/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7290 -
accuracy: 0.7401 - val_loss: 1.2122 - val_accuracy: 0.6048
Epoch 158/160
391/391 [==============================] - 101s 257ms/step - loss: 0.7255 -
accuracy: 0.7384 - val_loss: 1.2124 - val_accuracy: 0.6047
Epoch 159/160
391/391 [==============================] - 101s 258ms/step - loss: 0.7288 -
accuracy: 0.7396 - val_loss: 1.2126 - val_accuracy: 0.6052
Epoch 160/160
391/391 [==============================] - 100s 257ms/step - loss: 0.7271 -
accuracy: 0.7409 - val_loss: 1.2126 - val_accuracy: 0.6049
```

[ ]: 0.0

```
[ ]: vit.save_weights("./ViTAdam.h5")
```

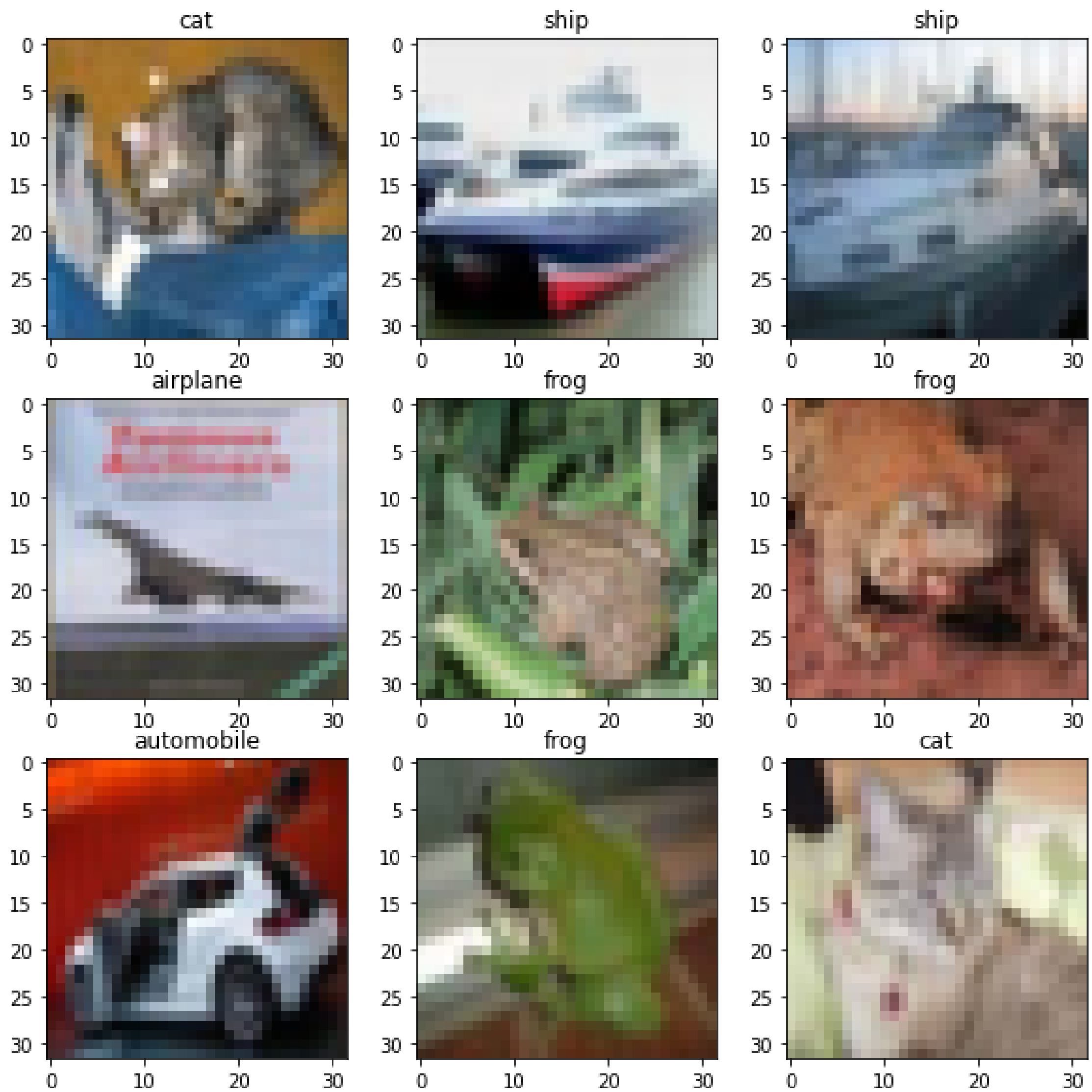
```
[ ]: vit.optimizer.lr.numpy()
```

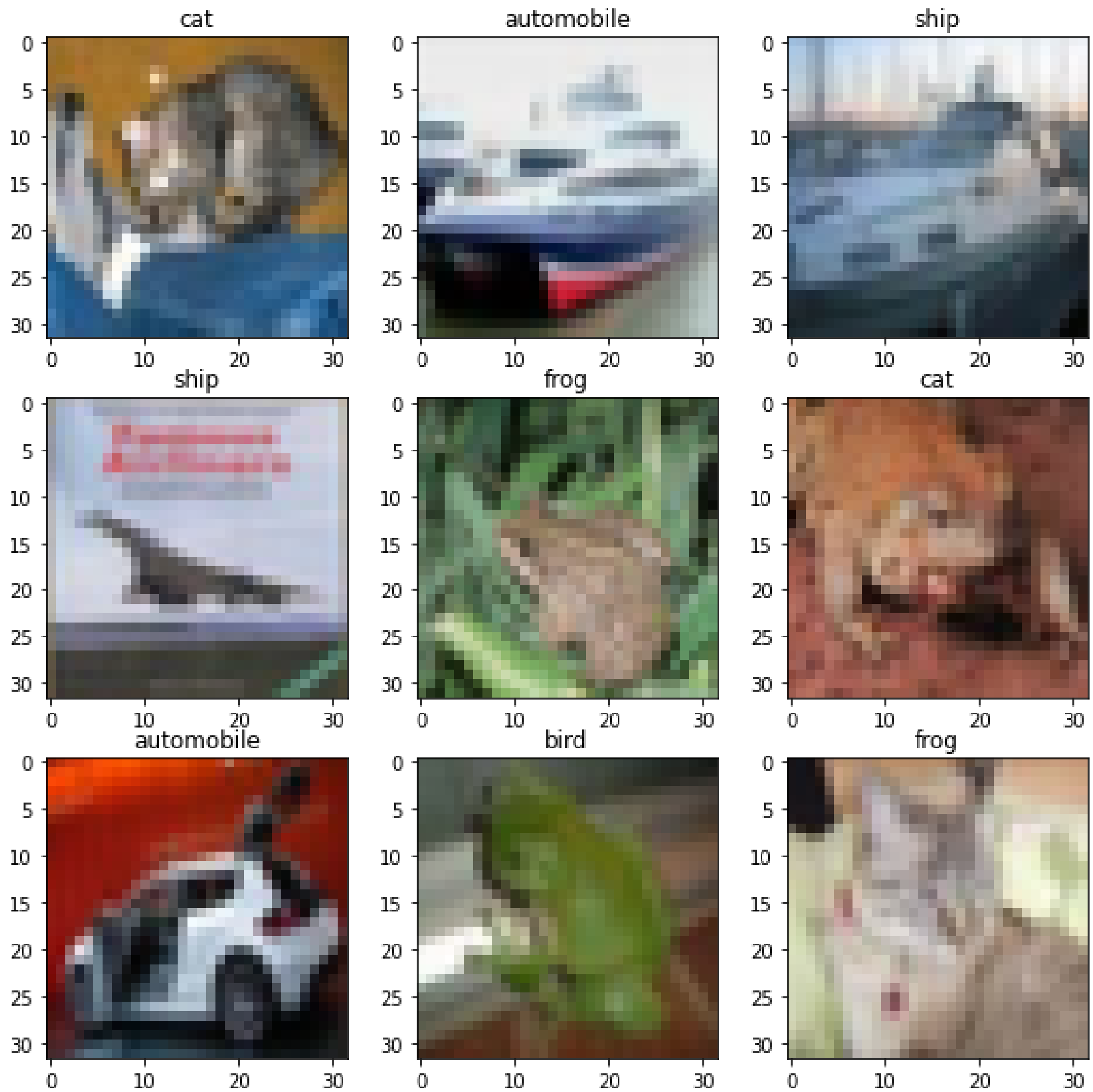
[ ]: 1e-07

Co ciekawe learning rate na koniec wynosi 1e-07, zaczynaliśmy od 1e-05, więc scheduler zadziałał dobrze (chociaż myślałem, że adam sam dostosowuje learning rate trochę bardziej?).
Model nauczył się całkiem nieźle, choć niestety dalej nie jakoś wybitnie i na zbiorze testowym

osiąga aż/tylko 60% accuracy.

```
[ ]: def peak_ds(ds, labels):
         plt.figure(figsize=(10,10))
         for ix, image in enumerate(test_images[:9]):
             plt.subplot(3,3, ix+1)
             plt.imshow(image)
             plt.title(classes[tf.math.argmax(labels[ix])])
         plt.show()

     peak_ds(test_images[:9], test_labels[:9])
     peak_ds(test_images[:9], vit.predict(test_images[:9]))
```





Jak zobaczymy co zwraca nasz model to można się nawet troszkę zaśmiać :-) aczkolwiek niektóre wyniki są dobre.

Próbowałem zrobić Attention Rollout, ale niestety przez to w jaki sposób zdefiniowałem model - przy użyciu functional API, i class'y Layer stworzyłem warstwy transformer i nie umiem z niej wyciągnąc informacji o output'cie multihead attention...
Nie mam pomysłu jak to łatwo poprawić, a nie mam też czasu uczyć nową sieć.
Z tego co rozumiem powininenem zaaplikować jakąś funkcję (mean, min) na output'cie multihead attention, tak żeby zebrać razem outputy wszystkich "tokenów" oraz tego trochę nie rozumiem, ale dodajemy macierz identyczności, żeby zasymulować skip connection?. Następnie takie outputy mnożymy między siebie.

Następnie spróbuję wykorzystać AdamW jako optimizer.
(Tak naprawdę najpierw przetestowałem AdamW i wyszedł gorzej, ale nie mam czasu szukać lep-

szych parametrów i zostawiłem AdamW na koniec)

[ ]:
```
optimizer = tfa.optimizers.AdamW(
    learning_rate=0.00001,
    weight_decay=0.00004
)
# optimizer = tf.keras.optimizers.Adam(learning_rate=0.00001)
vit.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.CategoricalCrossentropy(),
    metrics=['accuracy']
)
round(vit.optimizer.lr.numpy(), 5)
```

[ ]: 1e-05

[ ]:
```
history = vit.fit(
    x=train_images,
    y=train_labels,
    batch_size=128,
    epochs=160,
    callbacks=[cb],
    validation_data=(test_images, test_labels)
)
round(vit.optimizer.lr.numpy(), 5)
```

```
Epoch 1/160

2022-12-01 18:51:07.180023: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369]
Loaded cuDNN version 8005

391/391 [==============================] - 110s 262ms/step - loss: 2.3118 -
accuracy: 0.1584 - val_loss: 2.0603 - val_accuracy: 0.2379
Epoch 2/160
391/391 [==============================] - 101s 259ms/step - loss: 2.0671 -
accuracy: 0.2216 - val_loss: 1.9248 - val_accuracy: 0.2792
Epoch 3/160
391/391 [==============================] - 101s 260ms/step - loss: 1.9531 -
accuracy: 0.2556 - val_loss: 1.8527 - val_accuracy: 0.3038
Epoch 4/160
391/391 [==============================] - 101s 260ms/step - loss: 1.8719 -
accuracy: 0.2837 - val_loss: 1.8024 - val_accuracy: 0.3224
Epoch 5/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8198 -
accuracy: 0.3033 - val_loss: 1.7838 - val_accuracy: 0.3297
Epoch 6/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7735 -
accuracy: 0.3261 - val_loss: 1.7145 - val_accuracy: 0.3595
Epoch 7/160
```

```
391/391 [==============================] - 102s 260ms/step - loss: 1.7342 -
accuracy: 0.3451 - val_loss: 1.6734 - val_accuracy: 0.3865
Epoch 8/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6926 -
accuracy: 0.3650 - val_loss: 1.6385 - val_accuracy: 0.4002
Epoch 9/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6508 -
accuracy: 0.3865 - val_loss: 1.6215 - val_accuracy: 0.4019
Epoch 10/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6114 -
accuracy: 0.4062 - val_loss: 1.5779 - val_accuracy: 0.4214
Epoch 11/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5627 -
accuracy: 0.4255 - val_loss: 1.5198 - val_accuracy: 0.4455
Epoch 12/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5210 -
accuracy: 0.4446 - val_loss: 1.4730 - val_accuracy: 0.4652
Epoch 13/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4819 -
accuracy: 0.4612 - val_loss: 1.4510 - val_accuracy: 0.4717
Epoch 14/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4493 -
accuracy: 0.4733 - val_loss: 1.4239 - val_accuracy: 0.4843
Epoch 15/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4197 -
accuracy: 0.4882 - val_loss: 1.4184 - val_accuracy: 0.4823
Epoch 16/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3976 -
accuracy: 0.4943 - val_loss: 1.3853 - val_accuracy: 0.5033
Epoch 17/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3775 -
accuracy: 0.5018 - val_loss: 1.3794 - val_accuracy: 0.5086
Epoch 18/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3615 -
accuracy: 0.5120 - val_loss: 1.3577 - val_accuracy: 0.5186
Epoch 19/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3454 -
accuracy: 0.5187 - val_loss: 1.3498 - val_accuracy: 0.5178
Epoch 20/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3335 -
accuracy: 0.5230 - val_loss: 1.3500 - val_accuracy: 0.5171
Epoch 21/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3222 -
accuracy: 0.5257 - val_loss: 1.3520 - val_accuracy: 0.5195
Epoch 22/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3119 -
accuracy: 0.5316 - val_loss: 1.3214 - val_accuracy: 0.5320
Epoch 23/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.3007 -
accuracy: 0.5340 - val_loss: 1.3351 - val_accuracy: 0.5259
Epoch 24/160
391/391 [==============================] - 105s 267ms/step - loss: 1.2910 -
accuracy: 0.5356 - val_loss: 1.3181 - val_accuracy: 0.5315
Epoch 25/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2876 -
accuracy: 0.5380 - val_loss: 1.3314 - val_accuracy: 0.5285
Epoch 26/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2778 -
accuracy: 0.5427 - val_loss: 1.3111 - val_accuracy: 0.5370
Epoch 27/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2708 -
accuracy: 0.5458 - val_loss: 1.3203 - val_accuracy: 0.5354
Epoch 28/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2629 -
accuracy: 0.5500 - val_loss: 1.2969 - val_accuracy: 0.5450
Epoch 29/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2615 -
accuracy: 0.5500 - val_loss: 1.3068 - val_accuracy: 0.5379
Epoch 30/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2551 -
accuracy: 0.5535 - val_loss: 1.2970 - val_accuracy: 0.5413
Epoch 31/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2492 -
accuracy: 0.5538 - val_loss: 1.3094 - val_accuracy: 0.5394
Epoch 32/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2444 -
accuracy: 0.5561 - val_loss: 1.3048 - val_accuracy: 0.5366
Epoch 33/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2370 -
accuracy: 0.5605 - val_loss: 1.2850 - val_accuracy: 0.5495
Epoch 34/160
391/391 [==============================] - 101s 260ms/step - loss: 1.2351 -
accuracy: 0.5610 - val_loss: 1.3065 - val_accuracy: 0.5410
Epoch 35/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2325 -
accuracy: 0.5639 - val_loss: 1.3028 - val_accuracy: 0.5433
Epoch 36/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2272 -
accuracy: 0.5649 - val_loss: 1.2945 - val_accuracy: 0.5451
Epoch 37/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2272 -
accuracy: 0.5648 - val_loss: 1.2841 - val_accuracy: 0.5462
Epoch 38/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2235 -
accuracy: 0.5663 - val_loss: 1.2758 - val_accuracy: 0.5517
Epoch 39/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.2230 -
accuracy: 0.5664 - val_loss: 1.2807 - val_accuracy: 0.5498
Epoch 40/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2173 -
accuracy: 0.5683 - val_loss: 1.3007 - val_accuracy: 0.5432
Epoch 41/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2144 -
accuracy: 0.5720 - val_loss: 1.2779 - val_accuracy: 0.5525
Epoch 42/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2130 -
accuracy: 0.5725 - val_loss: 1.2786 - val_accuracy: 0.5509
Epoch 43/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2112 -
accuracy: 0.5744 - val_loss: 1.2711 - val_accuracy: 0.5566
Epoch 44/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2116 -
accuracy: 0.5733 - val_loss: 1.2683 - val_accuracy: 0.5534
Epoch 45/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2108 -
accuracy: 0.5744 - val_loss: 1.2754 - val_accuracy: 0.5501
Epoch 46/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2089 -
accuracy: 0.5749 - val_loss: 1.2905 - val_accuracy: 0.5488
Epoch 47/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2038 -
accuracy: 0.5799 - val_loss: 1.2913 - val_accuracy: 0.5448
Epoch 48/160
391/391 [==============================] - 102s 260ms/step - loss: 1.2040 -
accuracy: 0.5780 - val_loss: 1.2606 - val_accuracy: 0.5542
Epoch 49/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2042 -
accuracy: 0.5791 - val_loss: 1.2589 - val_accuracy: 0.5574
Epoch 50/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2044 -
accuracy: 0.5796 - val_loss: 1.2813 - val_accuracy: 0.5473
Epoch 51/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2017 -
accuracy: 0.5803 - val_loss: 1.2654 - val_accuracy: 0.5544
Epoch 52/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2003 -
accuracy: 0.5791 - val_loss: 1.2607 - val_accuracy: 0.5568
Epoch 53/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1992 -
accuracy: 0.5801 - val_loss: 1.2784 - val_accuracy: 0.5526
Epoch 54/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2006 -
accuracy: 0.5789 - val_loss: 1.2518 - val_accuracy: 0.5623
Epoch 55/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.2012 -
accuracy: 0.5801 - val_loss: 1.2543 - val_accuracy: 0.5587
Epoch 56/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1997 -
accuracy: 0.5830 - val_loss: 1.2691 - val_accuracy: 0.5546
Epoch 57/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1990 -
accuracy: 0.5816 - val_loss: 1.2649 - val_accuracy: 0.5523
Epoch 58/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1983 -
accuracy: 0.5856 - val_loss: 1.2554 - val_accuracy: 0.5580
Epoch 59/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1990 -
accuracy: 0.5847 - val_loss: 1.2544 - val_accuracy: 0.5588
Epoch 60/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1958 -
accuracy: 0.5845 - val_loss: 1.2607 - val_accuracy: 0.5552
Epoch 61/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1958 -
accuracy: 0.5856 - val_loss: 1.2650 - val_accuracy: 0.5535
Epoch 62/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1978 -
accuracy: 0.5839 - val_loss: 1.2557 - val_accuracy: 0.5521
Epoch 63/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1939 -
accuracy: 0.5879 - val_loss: 1.2759 - val_accuracy: 0.5538
Epoch 64/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1958 -
accuracy: 0.5865 - val_loss: 1.2460 - val_accuracy: 0.5645
Epoch 65/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1953 -
accuracy: 0.5871 - val_loss: 1.2598 - val_accuracy: 0.5534
Epoch 66/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1966 -
accuracy: 0.5844 - val_loss: 1.2785 - val_accuracy: 0.5519
Epoch 67/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1970 -
accuracy: 0.5854 - val_loss: 1.2596 - val_accuracy: 0.5565
Epoch 68/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1929 -
accuracy: 0.5873 - val_loss: 1.2440 - val_accuracy: 0.5640
Epoch 69/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1948 -
accuracy: 0.5859 - val_loss: 1.2501 - val_accuracy: 0.5621
Epoch 70/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1942 -
accuracy: 0.5870 - val_loss: 1.2593 - val_accuracy: 0.5517
Epoch 71/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.1945 -
accuracy: 0.5870 - val_loss: 1.2531 - val_accuracy: 0.5547
Epoch 72/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1935 -
accuracy: 0.5879 - val_loss: 1.2635 - val_accuracy: 0.5527
Epoch 73/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1958 -
accuracy: 0.5872 - val_loss: 1.2576 - val_accuracy: 0.5580
Epoch 74/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1940 -
accuracy: 0.5880 - val_loss: 1.2639 - val_accuracy: 0.5542
Epoch 75/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1930 -
accuracy: 0.5907 - val_loss: 1.2553 - val_accuracy: 0.5596
Epoch 76/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1941 -
accuracy: 0.5880 - val_loss: 1.2839 - val_accuracy: 0.5474
Epoch 77/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1953 -
accuracy: 0.5872 - val_loss: 1.2487 - val_accuracy: 0.5588
Epoch 78/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1940 -
accuracy: 0.5891 - val_loss: 1.3043 - val_accuracy: 0.5341
Epoch 79/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1938 -
accuracy: 0.5884 - val_loss: 1.2687 - val_accuracy: 0.5502
Epoch 80/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1947 -
accuracy: 0.5909 - val_loss: 1.2759 - val_accuracy: 0.5434
Epoch 81/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1921 -
accuracy: 0.5899 - val_loss: 1.3024 - val_accuracy: 0.5359
Epoch 82/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1931 -
accuracy: 0.5907 - val_loss: 1.2653 - val_accuracy: 0.5503
Epoch 83/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1947 -
accuracy: 0.5887 - val_loss: 1.2612 - val_accuracy: 0.5475
Epoch 84/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1929 -
accuracy: 0.5913 - val_loss: 1.2668 - val_accuracy: 0.5530
Epoch 85/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1926 -
accuracy: 0.5921 - val_loss: 1.2618 - val_accuracy: 0.5560
Epoch 86/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1944 -
accuracy: 0.5888 - val_loss: 1.2432 - val_accuracy: 0.5655
Epoch 87/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.1922 -
accuracy: 0.5900 - val_loss: 1.2538 - val_accuracy: 0.5576
Epoch 88/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1935 -
accuracy: 0.5906 - val_loss: 1.2427 - val_accuracy: 0.5657
Epoch 89/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1923 -
accuracy: 0.5899 - val_loss: 1.2390 - val_accuracy: 0.5664
Epoch 90/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1886 -
accuracy: 0.5895 - val_loss: 1.2578 - val_accuracy: 0.5535
Epoch 91/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1937 -
accuracy: 0.5902 - val_loss: 1.2474 - val_accuracy: 0.5592
Epoch 92/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1913 -
accuracy: 0.5920 - val_loss: 1.2507 - val_accuracy: 0.5572
Epoch 93/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1921 -
accuracy: 0.5902 - val_loss: 1.2568 - val_accuracy: 0.5542
Epoch 94/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1909 -
accuracy: 0.5922 - val_loss: 1.2408 - val_accuracy: 0.5597
Epoch 95/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1935 -
accuracy: 0.5923 - val_loss: 1.2505 - val_accuracy: 0.5590
Epoch 96/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1895 -
accuracy: 0.5903 - val_loss: 1.2377 - val_accuracy: 0.5663
Epoch 97/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1899 -
accuracy: 0.5920 - val_loss: 1.2530 - val_accuracy: 0.5563
Epoch 98/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1929 -
accuracy: 0.5908 - val_loss: 1.2647 - val_accuracy: 0.5547
Epoch 99/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1929 -
accuracy: 0.5900 - val_loss: 1.2413 - val_accuracy: 0.5625
Epoch 100/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1953 -
accuracy: 0.5899 - val_loss: 1.2585 - val_accuracy: 0.5500
Epoch 101/160
391/391 [==============================] - 101s 259ms/step - loss: 1.1792 -
accuracy: 0.6028 - val_loss: 1.2432 - val_accuracy: 0.5651
Epoch 102/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2032 -
accuracy: 0.5988 - val_loss: 1.2631 - val_accuracy: 0.5563
Epoch 103/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.2340 -
accuracy: 0.5942 - val_loss: 1.2777 - val_accuracy: 0.5547
Epoch 104/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2648 -
accuracy: 0.5888 - val_loss: 1.2978 - val_accuracy: 0.5486
Epoch 105/160
391/391 [==============================] - 101s 259ms/step - loss: 1.2961 -
accuracy: 0.5838 - val_loss: 1.3207 - val_accuracy: 0.5445
Epoch 106/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3270 -
accuracy: 0.5783 - val_loss: 1.3454 - val_accuracy: 0.5368
Epoch 107/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3592 -
accuracy: 0.5696 - val_loss: 1.3610 - val_accuracy: 0.5344
Epoch 108/160
391/391 [==============================] - 101s 259ms/step - loss: 1.3896 -
accuracy: 0.5630 - val_loss: 1.3951 - val_accuracy: 0.5236
Epoch 109/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4195 -
accuracy: 0.5555 - val_loss: 1.4126 - val_accuracy: 0.5210
Epoch 110/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4453 -
accuracy: 0.5468 - val_loss: 1.4330 - val_accuracy: 0.5178
Epoch 111/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4718 -
accuracy: 0.5402 - val_loss: 1.4542 - val_accuracy: 0.5112
Epoch 112/160
391/391 [==============================] - 101s 259ms/step - loss: 1.4968 -
accuracy: 0.5355 - val_loss: 1.4777 - val_accuracy: 0.5056
Epoch 113/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5203 -
accuracy: 0.5249 - val_loss: 1.5005 - val_accuracy: 0.4990
Epoch 114/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5416 -
accuracy: 0.5205 - val_loss: 1.5165 - val_accuracy: 0.4943
Epoch 115/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5629 -
accuracy: 0.5127 - val_loss: 1.5377 - val_accuracy: 0.4901
Epoch 116/160
391/391 [==============================] - 101s 259ms/step - loss: 1.5823 -
accuracy: 0.5049 - val_loss: 1.5546 - val_accuracy: 0.4813
Epoch 117/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6008 -
accuracy: 0.5004 - val_loss: 1.5768 - val_accuracy: 0.4748
Epoch 118/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6193 -
accuracy: 0.4928 - val_loss: 1.5985 - val_accuracy: 0.4702
Epoch 119/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.6354 -
accuracy: 0.4873 - val_loss: 1.6092 - val_accuracy: 0.4675
Epoch 120/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6514 -
accuracy: 0.4817 - val_loss: 1.6254 - val_accuracy: 0.4617
Epoch 121/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6667 -
accuracy: 0.4764 - val_loss: 1.6410 - val_accuracy: 0.4570
Epoch 122/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6806 -
accuracy: 0.4739 - val_loss: 1.6541 - val_accuracy: 0.4509
Epoch 123/160
391/391 [==============================] - 101s 259ms/step - loss: 1.6942 -
accuracy: 0.4664 - val_loss: 1.6733 - val_accuracy: 0.4467
Epoch 124/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7070 -
accuracy: 0.4620 - val_loss: 1.6765 - val_accuracy: 0.4462
Epoch 125/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7194 -
accuracy: 0.4562 - val_loss: 1.7004 - val_accuracy: 0.4323
Epoch 126/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7309 -
accuracy: 0.4524 - val_loss: 1.7071 - val_accuracy: 0.4332
Epoch 127/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7427 -
accuracy: 0.4489 - val_loss: 1.7145 - val_accuracy: 0.4337
Epoch 128/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7535 -
accuracy: 0.4436 - val_loss: 1.7340 - val_accuracy: 0.4226
Epoch 129/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7640 -
accuracy: 0.4377 - val_loss: 1.7455 - val_accuracy: 0.4138
Epoch 130/160
391/391 [==============================] - 105s 267ms/step - loss: 1.7740 -
accuracy: 0.4343 - val_loss: 1.7567 - val_accuracy: 0.4074
Epoch 131/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7830 -
accuracy: 0.4313 - val_loss: 1.7622 - val_accuracy: 0.4101
Epoch 132/160
391/391 [==============================] - 101s 259ms/step - loss: 1.7935 -
accuracy: 0.4264 - val_loss: 1.7777 - val_accuracy: 0.3998
Epoch 133/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8029 -
accuracy: 0.4223 - val_loss: 1.7781 - val_accuracy: 0.4082
Epoch 134/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8113 -
accuracy: 0.4177 - val_loss: 1.7911 - val_accuracy: 0.4062
Epoch 135/160
```

```
391/391 [==============================] - 101s 259ms/step - loss: 1.8200 -
accuracy: 0.4164 - val_loss: 1.7947 - val_accuracy: 0.4033
Epoch 136/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8286 -
accuracy: 0.4098 - val_loss: 1.8071 - val_accuracy: 0.3984
Epoch 137/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8365 -
accuracy: 0.4101 - val_loss: 1.8116 - val_accuracy: 0.3999
Epoch 138/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8447 -
accuracy: 0.4050 - val_loss: 1.8235 - val_accuracy: 0.3904
Epoch 139/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8524 -
accuracy: 0.4023 - val_loss: 1.8273 - val_accuracy: 0.3898
Epoch 140/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8600 -
accuracy: 0.3998 - val_loss: 1.8364 - val_accuracy: 0.3910
Epoch 141/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8677 -
accuracy: 0.3947 - val_loss: 1.8435 - val_accuracy: 0.3865
Epoch 142/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8751 -
accuracy: 0.3937 - val_loss: 1.8555 - val_accuracy: 0.3734
Epoch 143/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8823 -
accuracy: 0.3890 - val_loss: 1.8580 - val_accuracy: 0.3822
Epoch 144/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8894 -
accuracy: 0.3874 - val_loss: 1.8652 - val_accuracy: 0.3803
Epoch 145/160
391/391 [==============================] - 101s 259ms/step - loss: 1.8963 -
accuracy: 0.3857 - val_loss: 1.8751 - val_accuracy: 0.3728
Epoch 146/160
391/391 [==============================] - 101s 259ms/step - loss: 1.9033 -
accuracy: 0.3810 - val_loss: 1.8828 - val_accuracy: 0.3694
Epoch 147/160
391/391 [==============================] - 101s 259ms/step - loss: 1.9093 -
accuracy: 0.3787 - val_loss: 1.8919 - val_accuracy: 0.3615
Epoch 148/160
341/391 [=========================>…] - ETA: 12s - loss: 1.9164 - accuracy:
0.3742
```

Niestety obliczenia zawiesiły się przy 148 epokach i zawiesiły notebook, ale widać że być może parametry nie były dobrane najlepiej, ponieważ najlepsze accuracy jest w okolicach 0.55 przy 50 epoce, a później zaczęło spadać. Niestety nie mam czasu próbować szukać innych parametrów.

```
[ ]:
```