

Outline

Deep Learning

(520K0220)

Introduction

Residera

(Overfit)

ResNet50

CNN

CNN (Overfit)

(Vanishing)

Overall results

End

Deep Learning Midterm Assignment

P.L.D. Tien (520K0220)

Ton Duc Thang University

March 10, 2023



Outline

Deep Learning

P.L.D. Tien (520K0220)

Introduction

Overviev

ResNet5

ResNet5 (Overfit)

ResNet50 (Vanishing

IND

CNN (Overfi

CNN (Vanishir

Overall result

1 Introduction

2 Overview

3 ResNet50

4 ResNet50 (Overfit)

5 ResNet50 (Vanishing)

6 CNN

7 CNN (Overfit)

8 CNN (Vanishing)

9 Overall results



What is this?

Deep Learning

P.L.D. Tiei (520K0220

Introduction

Overvie

RecNets

ResNet5

ResNet50 (Vanishing

CNI

CNN (Overfit

CNN

Overall result

End

An image captioning system, using ResNet50 and CNN, with independent handling of overfitting and vanishing problems. This submission uses the CIFAR- 10^1 for it's dataset.

Note

The snippets of code are coming from the original notebook. It's advised to look at it for a better understanding.

¹https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz



What does it cover?

Deep Learning

(520K0220)

Introductio

Overview

ResNetb

ResNet5(Overfit)

ResNet50 (Vanishing

CNIN

CNN (Overfi

CNN

Overall result

- ResNet50
- ResNet50 (w/ Overfit handling)
- ResNet50 (w/Vanishing handling)
- CNN
- CNN (w/ Overfit handling)
- CNN (w/ Vanishing handling)



Implementation

Deep Learning

P.L.D. Tien (520K0220)

Introductio

0.00.00

ResNet50

(Overfit)

ResNet50

CNI

CNN (Overfit

CNN (Vanishing

Overall results

End

Intentionally left blank.

Too much to cover here, read the notebook.



Deep Learning

P.L.D. Tier (520K0220)

Introduction

ResNet50

ResNet5

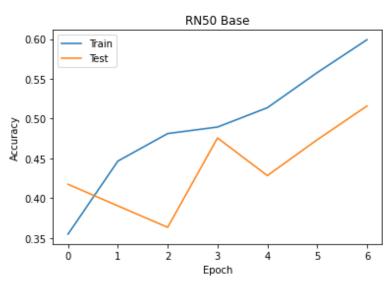
ResNet50

CNIN

CNN (Overfi

CNN

Overall results





Overview

Deep Learning

P.L.D. Tier (520K0220)

Introductio

ResNet5

ResNet50 (Overfit)

ResNet50

· _____

CNN (Overfit

CNN

Overall result

End

```
X = AveragePooling2D(pool_size=(2, 2), padding='same')(X)
        X = Flatten()(X)
40
41
        #X = Dense(256, activation='relu', name='fc1', kernel initializer='glorot uniform')(X)
        #X = Dense(128, activation='relu', name='fc2', kernel initializer='glorot uniform')(X)
42
43
        X = Dense(256, activation='relu', name='fc1', kernel_initializer='glorot_uniform'\
44
                  , kernel_regularizer=regularizers.l2(0.01))(X)
45
46
        X = Dense(128, activation='relu', name='fc2', kernel_initializer='glorot_uniform'\
     kernel_regularizer=regularizers.l2(0.01))(X)
47
```

Replace these lines with regularizers. Dropout doesn't work here since they're not tensors.



Deep Learning

P.L.D. Tien (520K0220)

Introduction

D. N.

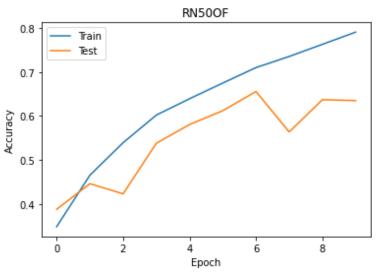
ResNet50 (Overfit)

ResNet50

CNN (Overfit)

CNN

Overall results





Implementation

Deep Learning

```
P.L.D. Tier
(520K0220)
```

Introduction

Overview

ResNet5

ResNet5

ResNet50 (Vanishing)

(vailisiiii

CNN (Overf

CNN (Vanishing)

Overall result

End

```
# add batch normalization after ReLU activation
         X = convolutional block(X, f=3, filters=[64, 64, 256], stage=2, block='a', stride=1)
         X = BatchNormalization(axis=3, name='bn_conv2')(X)
18
         X = ReLU()(X)
20
         X = identity_block(X, 3, [64, 64, 256], stage=2, block='b')
         X = BatchNormalization(axis=3, name='bn conv3')(X)
         X = ReLU()(X)
         X = identity block(X, 3, [64, 64, 256], stage=2, block='c')
24
         X = BatchNormalization(axis=3, name='bn conv4')(X)
         X = ReLU()(X)
26
         X = convolutional block(X, f=3, filters=[128, 128, 512], stage=3, block='a', stride=2)
28
         X = BatchNormalization(axis=3, name='bn conv5')(X)
         X = ReLU()(X)
30
         X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
31
         X = BatchNormalization(axis=3, name='bn conv6')(X)
         X = ReLU()(X)
         X = identity block(X, 3, [128, 128, 512], stage=3, block='c')
        X = BatchNormalization(axis=3, name='bn conv7')(X)
34
         X = ReLU()(X)
36
         X = identity block(X, 3, [128, 128, 512], stage=3, block='d')
```

We add BatchNormalization after ReLU.



Deep Learning

P.L.D. Tien (520K0220)

Introductio

DocMo+5

ResNet5

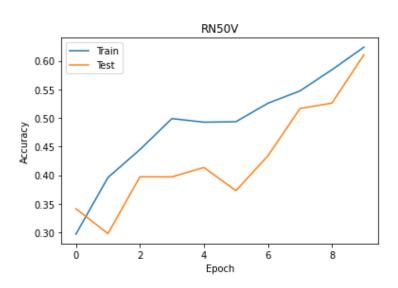
ResNet50 (Vanishing)

CNN

CNN (Overfi

CNN

Overall results





Overview

Deep Learning

P.L.D. Tien (520K0220)

Introductio

Overview

RecNet

ResNet5

ResNet50

(Vanishing

CNN

Civit (Overing

CNN (Vanishing)

Overall result

End

```
model = models.Sequential(
   layers.Conv2D(32, (3, 3), activation='relu', kernel initializer='he uniform', padding='same', input shape=(32, 32, 3)),
   layers.Conv2D(32, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
   layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same').
   layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
   layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
   layers.Conv2D(128, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
   layers.Conv2D(128, (3, 3), activation='relu', kernel initializer='he uniform', padding='same').
   layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
   layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
   layers.Conv2D(256, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
   layers.Conv2D(256, (3, 3), activation='relu', kernel initializer='he uniform', padding='same').
   layers.Flatten(),
   layers.Dense(64, activation='relu'),
   layers.BatchNormalization(),
   layers.Dense(10, activation='softmax')
```

Simple enough.



Deep Learning

P.L.D. Tien (520K0220)

Introduction

D N . F

ResNet5

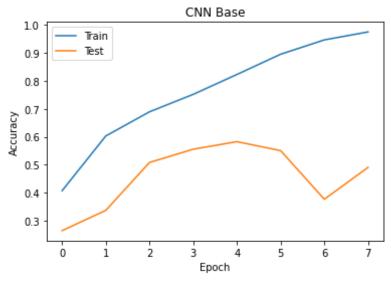
ResNet50

CNN

CNN (Overfi

CNN

Overall results





Overview

Deep Learning

P.L.D. Tien (520K0220)

Introduction

Overview

ResNetF

ResNet5

ResNet50

(vailisiiii

CNN (Overfit)

CNN

Overall result

End

```
layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', inp
layers.Conv2D(32, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
layers.BatchNormalization(),
layers.MaxPooling2D((2, 2)),
layers.Dropout(0.1).
layers.Conv2D(64, (3, 3), activation='relu', kernel initializer='he uniform', padding='same').
layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same').
layers.BatchNormalization().
layers.MaxPooling2D((2, 2)),
layers.Dropout(0.1),
layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.Conv2D(128, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
layers.BatchNormalization(),
layers.MaxPooling2D((2, 2)),
layers.Dropout(0.1),
layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.Conv2D(256, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
layers.BatchNormalization().
layers.MaxPooling2D((2, 2)),
layers.Dropout(0.1),
```

We simply add Dropout in between the blocks.



Deep Learning

P.L.D. Tien (520K0220)

Introduction

ResNet5

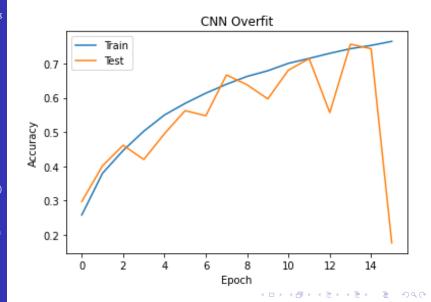
ResNet50

(Vanishing

CNN (Overfit)

CNN

Overall results





Implementation

Deep Learning

P.L.D. Tien (520K0220)

Introductio

Overview

DaaNlase

ResNet5

(Overfit)

(Vanishing

CNN

CNN (Overfi

CNN (Vanishing)

Overall results

End

```
layers.Conv2D(32, (3, 3), activation='relu', kernel initializer='he uniform', padding='same', input shape=(32, 32, 3)),
layers.BatchNormalization(),
layers.Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization().
layers.MaxPooling2D((2, 2)),
layers.Dropout(0.2),
layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers, BatchNormalization(),
layers.Conv2D(64, (3, 3), activation='relu', kernel initializer='he uniform', padding='same'),
layers.BatchNormalization(),
layers.Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization(),
layers.MaxPooling2D((2, 2)).
layers.Dropout(0.3),
layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform'. padding='same').
layers.BatchNormalization().
layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization(),
layers.Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization(),
layers.MaxPooling2D((2, 2)).
layers, Dropout(0,4),
layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization().
layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same').
layers.BatchNormalization(),
layers.Conv2D(256, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'),
layers.BatchNormalization(),
layers.MaxPooling2D((2, 2)).
layers, Dropout(0.5).
```

Add BatchNormalization in between Conv2D.



Deep Learning

P.L.D. Tien (520K0220)

Introduction

DanNate

ResNet5

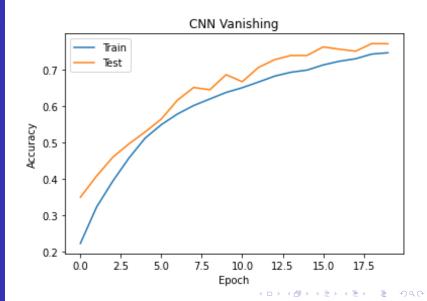
PocNotE0

(Vanishing

CNN (Over

CNN (Vanishing)

Overall results





All six of them together

Deep Learning

P.L.D. Tier (520K0220)

Introduction

. . . .

RecNet5

(Overfit)

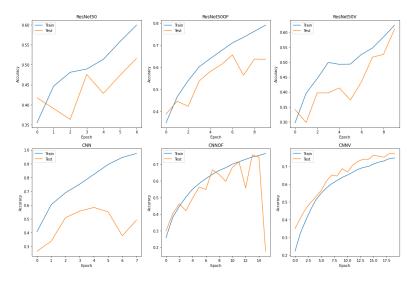
(Vanishing

IND

CNN (Overfi

CNN (Vanishing)

Overall results





Deep Learning

P.L.D. Tien (520K0220)

Introduction

Overviev

D. N. E

ResNet50

ResNet50

CRIBI

CNN (Vanishing

Overall results

End

Anything to talk about?