



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر

## **پروژه پنجم درس رایانش عصبی**

نگارش

سیدمهدی میرفندرسکی

مدرس

دکتر رضا صفابخش

دی ۱۴۰۱

## فهرست مطالب

۱	سوال اول	۲
۲	سوال دوم	۲
۳	سوال سوم	۵
۴	سوال چهارم	۶
۵	سوال پنجم	۶

## ۱ سوال اول

اساساً با این کار به نوعی می‌توانیم داده‌های سری زمانی را به یک مسئله یادگیری نظارت‌شده یا بدون نظارت تبدیل کنیم. در واقع در نظر گرفتن لگ‌ها در تجزیه و تحلیل سری‌های زمانی بسیار مفید هستند زیرا در بین داده‌های سری زمانی خودهمبستگی وجود دارد که همبستگی مقادیر درون یک سری زمانی با مقادیر قبلی خود است. البته بسته به مسئله هرچه از داده فعلی دور شویم، امکان دارد این همبستگی کم یا زیاد شود. با این کار در این مسئله، مشخص می‌کنیم که شبکه نسبت به چقدر داده‌های قبلی حساس باشد. به عنوان مثال احتمالاً در مسئله بورس بیشتر از شش ماه گذشته وابستگی نخواهند داشت. در ادامه برای حل این بخش تعداد گام‌های ورودی برابر با ۳۲ و همچنین با offset یک، پنجره‌ها ایجاد شدند. عدد ۳۲ بدین دلیل بود که تاریخچه نزدیک به یک ماه در نظر گرفته شود.

## ۲ سوال دوم

برای آموزش شبکه تلاش‌های بسیاری صورت گرفت اما نتایج قابل قبولی دریافت نشد. در نهایت معماری انتخاب شده بدین صورت است که یک لایه SimpleRNN، یک لایه Dense و در نهایت یک تک نورون برای مسئله دسته‌بندی در نظر گرفته شد. در ضمن تابع فعالیت خروجی شبکه‌ها سیگموئید و توابع فعلیت لایه‌های دیگر relu در نظر گرفته شد. همچنین سائز دسته برابر ۱۶ و ای‌پاک برابر با ۱۰ در نظر گرفته شد. مشابه تمرینات قبل کالک زیر نیز گذاشته شد.

`es-callback = EarlyStopping(monitor="val-loss", patience=3, restore-best-weights=True)`

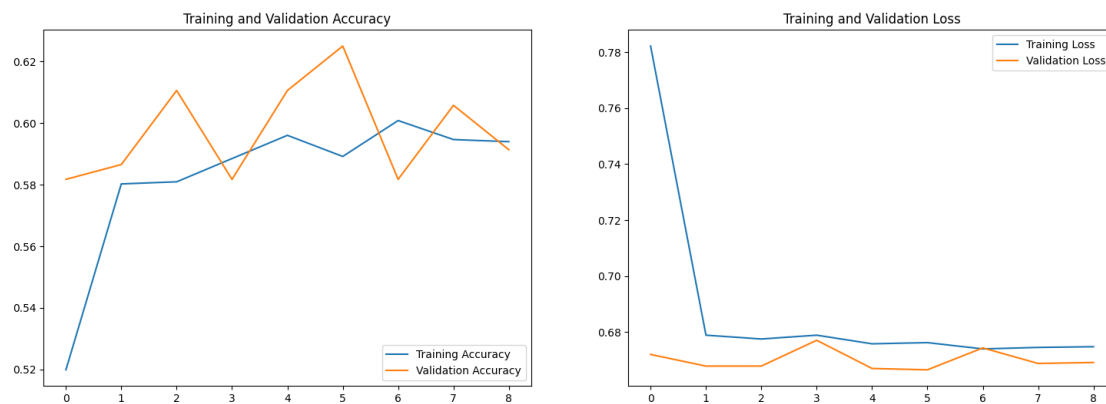
در جدول زیر ترکیبات مختلف تعداد واحدها مشاهده می‌شود.

شماره	SimpleRNN	Dense	صحت آموزشی	صحت اعتبارسنجی	صحت تست
۱	۱	۱۶	58.57	58.17	58.03
۲	۲	۱۶	58.57	58.17	58.03
۳	۳	۱۶	58.57	58.17	58.03
۴	۴	۱۶	56.52	62.5	57.79
۵	۵	۱۶	59.05	59.13	57.79
۶	۶	۱۶	58.57	58.17	58.03
۷	۷	۱۶	59.12	58.65	57.55
۸	۱	۳۲	58.57	58.17	58.03
۹	۲	۳۲	58.78	58.65	57.55
۱۰	۳	۳۲	58.85	58.65	58.03
۱۱	۴	۳۲	58.57	58.17	58.03
۱۲	۵	۳۲	59.05	61.54	61.15
۱۳	۶	۳۲	58.57	58.17	58.03
۱۴	۷	۳۲	58.57	58.17	58.03
۱۵	۱	۶۴	58.57	58.17	58.03
۱۶	۲	۶۴	58.5	62.02	59.71
۱۷	۳	۶۴	58.64	58.65	57.79
۱۸	۴	۶۴	58.85	58.65	57.55
۱۹	۵	۶۴	58.57	58.17	58.03
۲۰	۶	۶۴	58.98	57.69	58.99
۲۱	۷	۶۴	60.29	60.1	59.23
۲۲	۱	۱۲۸	58.57	58.17	58.03
۲۳	۲	۱۲۸	58.57	58.17	58.03
۲۴	۳	۱۲۸	58.57	58.17	58.03
۲۵	۴	۱۲۸	58.23	60.1	60.43
۲۶	۵	۱۲۸	57.89	60.58	60.19
۲۷	۶	۱۲۸	58.5	61.54	61.15
۲۸	۷	۱۲۸	57.61	62.5	58.03

با در نظر گرفتن صحت مجموعه آموزشی و اعتبارسنجی معماری زیر انتخاب شد:

صحت تست	صحت اعتبارسنجی	صحت آموزشی	Dense	SimpleRNN	شماره
59.71	62.02	58.5	۶۴	۲	۱۶

همچنین نمودار صحت و خطای مجموعه آموزشی و اعتبارسنجی برای بهترین معماری در ادامه مشاهده می‌شود.



شکل ۱: نمودار صحت و خطای بهترین معماری

### ۳ سوال سوم

برای این بخش یک مدل شبکه کانولوشنی آموزش دیده شد. توجه شود که برای این قسمت سعی و خطا در نظر گرفته نشده است (سعی و خطا شبکه کانولوشنی در پروژه چهارم). معماری شبکه طراحی شده و نتایج آن در ادامه مشاهده می‌شود.

Model: "sequential\_95"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 32, 2, 16)	80
average_pooling2d_2 (Average Pooling2D)	(None, 32, 1, 16)	0
Flat (Flatten)	(None, 512)	0
Dense (Dense)	(None, 64)	32832
Output (Dense)	(None, 1)	65

=====

Total params: 32,977  
Trainable params: 32,977  
Non-trainable params: 0

شکل ۲: معماری شبکه کانولوشنی



شکل ۳: نمودارهای شبکه کانولوشنی

همانطور که مشاهده می‌شود این شبکه کانولوشنی تقریباً در حد بهترین شبکه قسمت قبل عمل کرده است (می‌توان گفت کانولوشنی حتی مقداری بهتر عمل کرده است). همچنین احتمالاً با سعی و خطا بتوان نتایج بهتری را نیز کسب کرد. پس در کل اینگونه برداشت شد که شبکه کانولوشنی برای این مسئله بهتر عمل می‌کند.

## ۴ سوال چهارم

در این گزارش ابتدا انواع ناهنجاری‌ها بیان شده است: (داده‌های پرت افزایشی، تغییرات زمانی و تغییر سطح یک پدیده). در ادامه مجموعه داده قیمت بیتکوین برای ادامه کار در نظر گرفته شده است. سپس مجموعه آموزشی و تست را ایجاد و آن‌ها را با یک روش یکسان نرمالایز کرده است. در نهایت برای آخرین قسمت پیش‌پردازش مقدار لگ ۳۰ برای آموزش انتخاب شده است. سپس شبکه خود کدگذاری با معماری زیر شامل لایه کانولوشنی و LSTM طراحی کرده است. سپس به آموزش شبکه پرداخته شد (بوضوح ورودی و خروجی شبکه داده‌های آموزشی هستند). سپس مشابه تمرین گذاشته با نمودارهای خطای آموزش و اعتبارسنجی، بایاس و واریانس مدل را بررسی می‌کند. حال که مدل از نظر بایاس و واریانس مطلوب است، نوبت به محاسبه خطای بازسازی کل شبکه خود کدگذار می‌رسد. در این مرحله خطای بازسازی داده‌های آموزشی و تست محاسبه می‌شوند. در داده‌های تست اگر مقداری از حداکثر مقدار خطای آموزشی (mae) بیشتر باشد، در لیست ناهنجاری‌ها ذخیره می‌شود (اساس این روش بر این اساس است که اگر داده‌ای از تست بازسازی مناسبی نداشت به عنوان ناهنجاری شناخته شود).

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 30, 32)	512
LSTM_layer_1 (LSTM)	(None, 25)	5800
repeat_vector (RepeatVector)	(None, 30, 25)	0
LSTM_layer_2 (LSTM)	(None, 30, 25)	5100
conv1d_1 (Conv1D)	(None, 30, 32)	12032
time_distributed (TimeDistrib	(None, 30, 1)	33
Total params: 23,477		
Trainable params: 23,477		
Non-trainable params: 0		

شکل ۴: معماری سوال چهارم

## ۵ سوال پنجم

این مقاله از شبکه‌های خود کدگذار برای یادگیری خود نظارتی در مسائل بینایی ماشین استفاده کرده است. بدین صورت که ابتدا یک شبکه خود کدگذار نامتقارن با قسمت کدگذار با پارمترهای بیشتر در نظر گرفته است. سپس به صورت تصادفی بعضی پیکسل‌های ورودی را ماسک (صفر) کرده است. البته کار متمایزی که این مقاله انجام داده است این است که حدود ۷۵ درصد ورودی را ماسک کرده است و ۲۵ درصد باقی مانده را جهت آموزش به شبکه خودکدگذار وارد کرده است. که همین امر باعث افزایش سرعت شده است. و در نهایت وقتی آموزش شبکه به اتمام رسید. ماسک‌های تصاویر تنها به قسمت کدگشای شبکه داده می‌شود تا تصویر بازسازی شود. این ایده کمک بسیار زیادی به تکمیل تصاویر ناقص می‌کند. نکته قابل توجه این خواهد بود که داده‌های ماسک شده امکان ورود به کدگذار را نخواستند داشت و تنها باید به قسمت کدگشا وارد شود.