به نام خدا

دانشکده علوم و فنون نوین

مبانی محاسبات (رایانش) نرم (۰۱-۰۳۹-۰۵-۸۳)

نیم‌سال اول ۱۴۰۲-۱۴۰۱

تمرین شماره ۳

تاریخ تحویل: ۱۴۰۱/۱۰/۰۲

---

۱. (۲۵٪) [پژوهش و محاسبه] برای هر کدام از سوالات زیر پاسخ را بدست آورید.

۱-۱ [۵٪] دو مساله Vanishing problem و Exploiding problem را توضیح داده و راه‌حل‌هایی را برای برطرف کردن این مشکل‌ها پیشنهاد دهید. کدامیک از توابع فعالسازی می‌توانند باعث این دو پدیده شوند؟

۱-۲ [۵٪] در مورد L1 Regularization و L2 Regularization تحقیق کنید و تفاوت میان آن‌ها را با جزئیات شرح دهید و مشخص کنید هر کدام از آنها چگونه بر به‌روزرسانی وزن‌های مدل اثر می‌گذارند. کدام مورد را ترجیح می‌دهید؟ دلیل انتخاب خود را توضیح دهید.

۱-۳ [۵٪] پدیده شکست تقارن (Symmetry breaking) چیست؟ آن را توضیح دهید. برای جلوگیری از آن چه کار(های)ی می‌توان انجام داد؟

۱-۴ [۱۰٪] فرض کنید یک شبکه عصبی تک لایه برای طبقه‌بندی دودویی(Binary classification) دارید. ورودی این شبکه $X \in R^{n*m}$ و خروجی $\hat{y} \in R^{1*m}$ و برچسب واقعی $y \in R^{1*m}$ است. فرمول‌های پیش‌خور(Forward propagation) به صورت زیر هستند:

$$z^{[1]} = W^{[1]}X + b^{[1]}$$
$$a^{[1]} = \sigma(z^{[1]})$$
$$\hat{y} = a^{[1]}$$
$$J = -\sum_{i=1}^{m} y^{(i)} log(\hat{y}^{[i]}) + (1 - y^{(i)}) log(1 - \hat{y}^{[i]})$$

عبارت $\frac{\partial J}{\partial W^{[1]}}$ را با استفاده از قانون زنجیر (Chain rule) به دست آورید.

---

۲. (۷۵٪) [پیاده‌سازی: دسته‌بندی تصاویر با شبکه عصبی پیچشی] در این بخش قرار است که با استفاده از یک چارچوب برنامه‌نویسی یادگیری عمیق، یک مدل برای دسته‌بندی تصاویر مجموعه داده‌های CIFAR-10 پیاده‌سازی کنیم. این مجموعه داده از ۶۰ هزار تصویر رنگی با ابعاد ۳۲×۳۲ تشکیل شده است (شامل ۵۰ هزار تصویر آموزشی و ۱۰ هزار تصویر آزمایشی). تصاویر این مجموعه داده در ده دسته قرار گرفته‌اند که نمونه‌ای از هریک از این دسته‌ها در شکل ۱ ملاحظه می‌شود.

h.veisi@ut.ac.ir

به نام خدا

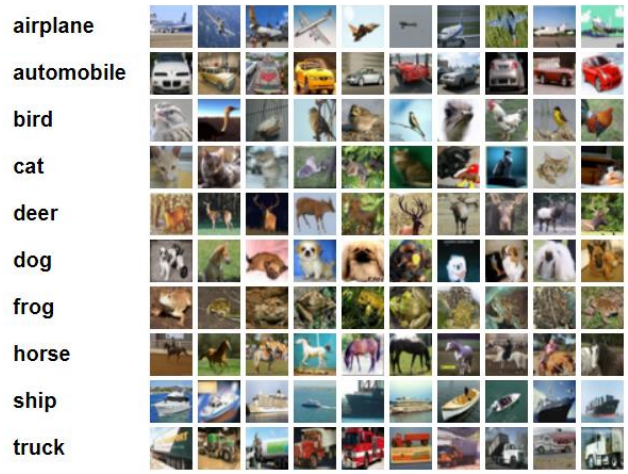
دانشکده علوم و فنون نوین

مبانی محاسبات (رایانش) نرم (۰۱-۰۳۹-۰۵-۸۳)

نیم‌سال اول ۱۴۰۲-۱۴۰۱

تمرین شماره ۳

تاریخ تحویل: ۱۴۰۱/۱۰/۰۲



شکل ۱ نمونه تصاویری از هریک از برچسب‌های مجموعه داده‌ی CIFAR-10

به صفحه‌ی رسمی این مجموعه داده[1] مراجعه کنید، آن را دریافت کرده و مجموعه‌های آزمایشی و آموزشی آن را در این بخش از تمرین مورد استفاده قرار دهید. داده‌های آموزشی این پایگاه داده در پنج دسته‌ی جداگانه (هریک با اندازه‌ی ۱۰ هزار تصویر) ارائه شده‌اند. در صورتی که منابع شما اجازه‌ی اجرای الگوریتم بر روی تمامی مجموعه داده را نمی‌دهد، می‌توانید به عنوان داده‌ی آموزشی، تنها دسته اول آن (data_batch_1) را مورد استفاده قرار دهید. توجه کنید که در این صورت لازم است این موضوع را در گزارش خود ذکر کنید.

شبکه‌ای پیچشی با معماری زیر در نظر بگیرید:

۱) لایه‌ی پیچشی: اندازه‌ی کرنل: ۳×۳ ، (تعداد کانال‌های خروجی: ۷ و تابع فعال‌سازی: ReLU)
۲) لایه‌ی پیچشی: اندازه‌ی کرنل: ۳×۳ ، (تعداد کانال‌های خروجی: ۹ و تابع فعال‌سازی: ReLU)
۳) لایه‌ی ادغام بیشینه: اندازه‌ی کرنل: ۲×۲
۴) لایه drop-out با احتمال ۳۰ درصد
۵) لایه خطی با تعداد ده (تعداد دسته‌ها) نرون خروجی

[1] https://www.cs.toronto.edu/~kriz/cifar.html

h.veisi@ut.ac.ir

این شبکه را با بهینه‌ساز Adam، تابع خطای categorical cross entropy و نرخ یادگیری یک صدم آموزش دهید (با اندازه دسته ۳۲). این معماری و نحوه‌ی آموزش را به عنوان حالت پایه در نظر گرفته و در هریک از بخش‌های **الف** تا **خ** تغییرات گفته شده را بر روی همین معماری و پارامترها اعمال کنید.

نکته ۱: قسمتی از داده‌های آموزشی را به عنوان داده‌های validation جدا کرده (می‌توان به عنوان یک پارامتر هنگام فراخوانی تابع آموزش نیز تنظیم شود) و میزان خطای آن را معیاری برای زمان توقف الگوریتم قرار دهید. همچنین از دقت داده‌های آزمایشی به منظور ارزیابی مدل استفاده کنید.

نکته ۲: برای هریک از بخش‌های **الف** تا **ح**، موارد زیر را گزارش کنید:

- نموداری از روند تغییرات خطای شبکه در حین آموزش، بر روی داده‌های train و validation (این دو نمودار را روی یکدیگر قرار دهید)
- دقت مدل بر روی داده‌های آزمایشی
- میانگین مدت زمان اجرای تکرار[2]ها
- تحلیل تغییرات «سرعت همگرایی بر اساس تعداد تکرار» و «توان شبکه در یادگیری الگو» در اجراهای مختلف
- بررسی تمام موارد بالا و تحلیل علت تغییرات آن‌ها

الف) تعداد لایه‌های پیچشی را از دو لایه به سه و چهار لایه (با اندازه کرنل ۳×۳) تغییر دهید.

ب) تغییر اندازه‌ی کرنل لایه اول پیچشی به ۵×۵ و ۷×۷

پ) استفاده از یک لایه‌ی پیچشی با اندازه‌ی کرنل ۵×۵ به جای استفاده از لایه‌های اول و دوم

ت) تغییر ضریب یادگیری بهینه‌ساز به 10^{-4}، 10^{-3} و 10^{-1}

ث) تغییر تابع فعال‌ساز به leaky ReLU و tanh

ج) تغییر بهینه‌ساز به SGD و Adam

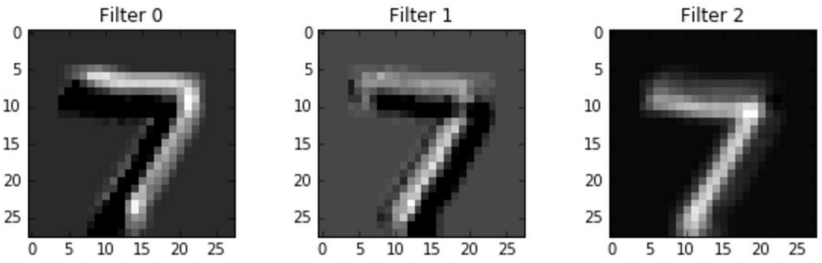
چ) اضافه کردن لایه‌ی Batch Normalization به عنوان لایه اول شبکه

ح) تغییر اندازه‌ی batch-size به ۴ و ۱۲۸

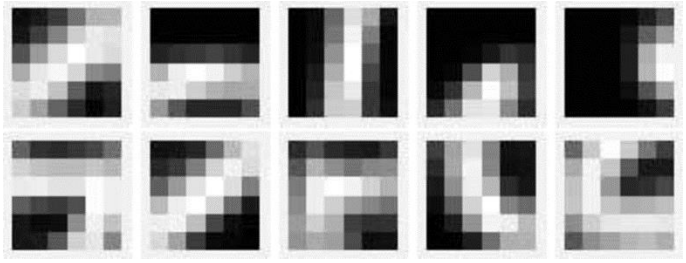
[2] epoch

h.veisi@ut.ac.ir

خ) پس از آموزش مدل، یکی از تصاویر آزمایشی را در نظر بگیرید، آن را نمایش دهید و آن را به عنوان ورودی به شبکه بدهید. پس از عبور تصویر از لایه‌ی اول پیچشی، هریک از کانال‌ها تصویری خاکستری [3] با ویژگی خاصی از تصویر ورودی را نمایش می‌دهد (شبیه به شکل ۲). همچنین وزن‌های مربوط به یکی از کانال‌های کرنل را نیز می‌توان به عنوان یک تصویر خاکستری در نظر گرفت و آن را نمایش داد (شبیه به شکل ۳). حال شما تصویر کانال‌های مختلف لایه اول پیچشی را در کنار تصویر وزن‌های کرنل‌های همان کانال نمایش دهید و ارتباط آن‌ها را توضیح دهید.



شکل ۲ نمونه سه تصویر که توسط سه فیلتر مختلف در لایه‌های میانی شبکه‌ی عصبی تولید شده است



شکل ۳ نمونه ۱۰ تصویر از وزن‌های کرنل (فیلتر) با اندازه‌ی ۶×۶

د) معماری MobileNetV2 به عنوان یک معماری شبکه‌ی عصبی سبک برای انجام کاربردهای تشخیص اشیا در تلفن‌های همراه معرفی شده است [4]. در این بخش، با به‌کار گیری نمونه‌ی

[3] gray-scale
[4] https://arxiv.org/abs/1801.04381

h.veisi@ut.ac.ir

پیش‌آموزش دیده‌شده‌ی این معماری بر روی مجموعه‌داده‌ی ImageNet[5] (که به صورت آماده در چارچوب‌های برنامه‌نویسی tensorflow و pytorch قرار داده‌شده است)، می‌خواهیم مدلی برای دسته‌بندی مجموعه‌ی داده‌ی CIFAR-10 آموزش دهیم. برای این منظور لازم است به انتهای بخش استخراج ویژگی[6] این شبکه، با وزن‌های آموزش دیده‌شده، لایه‌ای خطی (با تعداد خروجی ۱۰ عدد، تعداد برچسب‌ها) اضافه شود. سرعت همگرایی این مدل بر حسب تکرار و دقت آن بر روی داده‌های آزمایشی را با حالت پایه مقایسه، علت تفاوت‌ها را بیان کرده و نکات مثبت و منفی استفاده از مدل پیش‌آموزش دیده‌شده را بررسی کنید.

به عنوان مثال در نمونه کد زیر، استفاده از شبکه‌ی پیش‌آموزش دیده‌شده‌ی MibileNetV2 (بر روی پایگاه داده‌ی ImageNet) در چارچوب برنامه‌نویسی tensorflow آورده شده است که در آن تنظیم پارامتر include_top با مقدار False، موجب می‌شود که خروجی مدل بارگذاری شده، ویژگی‌های استخراج شده از تصاویر باشد (مدل بدون لایه/لایه‌های خطی انتهایی، بارگذاری می‌شود). در نهایت متغیر model شبکه‌ی مورد نظر در بخش **د** را شامل خواهد شد.

```
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2

pre_trained_model = MobileNetV2(weights='imagenet', include_top=False)
flatten_layer = tf.keras.layers.Flatten()
dense_layer = tf.keras.layers.Dense(10, activation='softmax')

input_images = tf.keras.Input(shape=(32, 32, 3), name='input_image')
features = pre_trained_model(input_images)
flatten_features = flatten_layer(features)
final_outputs = dense_layer(flatten_features)

model = tf.keras.Model(inputs=input_images, outputs=final_outputs)
```

[5] http://image-net.org/
[6] Feature extraction

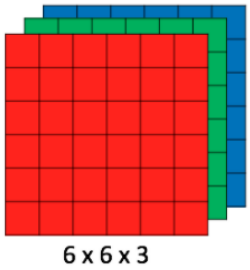
h.veisi@ut.ac.ir

به نام خدا

پیوست: مروری بر لایه‌ی پیچشی[7] در شبکه‌های عصبی عمیق

در شبکه‌های عصبی عمیق، لایه‌های پیچشی، نقشی کلیدی در بازشناسی الگو در تصاویر دارد. این لایه تقریباً در تمامی معماری‌های شبکه‌ی عصبی برای کاربردهای پردازش تصویر به کار می‌رود. حتی در کاربردهای غیر تصویری نیز گاهی پس از تبدیل داده‌ی ورودی به یک تصویر، از لایه‌ی پیچشی برای بازشناسی الگوی آن استفاده می‌شود. به عنوان مثال در کاربردهایی نظیر شناسایی اشیاء، تشخیص چهره و تشخیص گفتار (با تبدیل صوت به تصویر)، لایه‌ی پیچشی به طور گسترده مورد استفاده قرار می‌گیرد.

در کاربرد «دسته‌بندی تصاویر» یک تصویر ورودی، معمولا یک تصویر رنگی است به صورت یک ماتریس سه‌بعدی در قالبی مانند شکل ۴ ذخیره می‌شود. در بعد سوم این ماتریس، کانال‌های رنگی تصویر قرار می‌گیرد که به ترتیب شامل مقادیر روشنایی برای نورهای قرمز، سبز و آبی است.



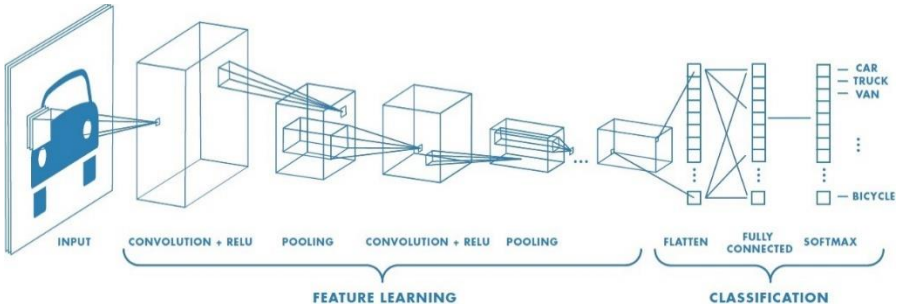
شکل ۴ نمایی از ماتریس سه‌بعدی یک تصویر رنگی با ابعاد ۶×۶

یک شبکه‌ی عصبی عمیق شامل لایه‌ی پیچشی، معمولا شامل لایه‌های ادغام و خطی (fully-connected) نیز می‌شود. در ادامه توضیح مختصری در مورد لایه‌ی پیچشی و ادغام آورده شده است. در شکل ۵ یک نمونه شبکه‌ی عصبی معمول که به وسیله‌ی آن دسته‌بندی تصاویر صورت می‌گیرد، ملاحظه می‌شود.

[7] Convolutional

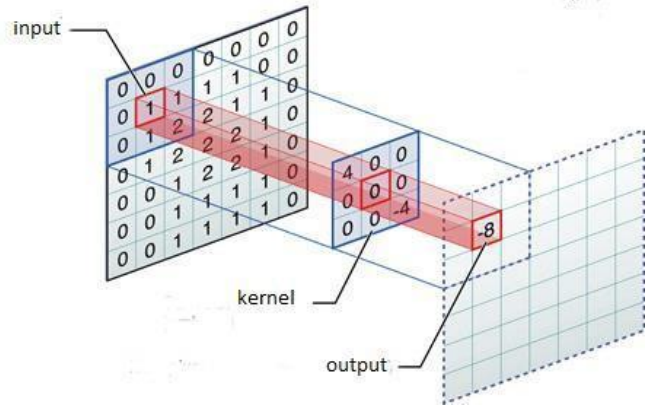
h.veisi@ut.ac.ir

به نام خدا



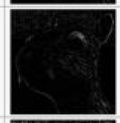
شکل ۵ نمونه معماری یک شبکه‌ی عصبی برای دسته‌بندی تصاویر

لایه‌ی پیچشی: معمولا به عنوان اولین لایه‌ها برای استخراج ویژگی از تصویر در نظر گرفته می‌شود. این لایه، در طول آموزش یاد می‌گیرد که چه ویژگی‌های محلی را از تصویر ورودی باید استخراج کند. تک تک پیکسل‌های خروجی این لایه با استفاده از یک فیلتر و حرکت کردن آن بر روی تصویر ورودی مانند شکل ۶ محاسبه می‌شود.



شکل ۶ نحوه‌ی اعمال فیلتر بر روی تصویر در لایه‌ی پیچشی

اعمال فیلترهای مختلف بر روی تصویر ورودی می‌تواند ویژگی‌های مختلفی از تصویر ورودی را استخراج کند. در شکل ۷ نمونه‌هایی از این ویژگی‌ها و فیلترهای متناظر آن‌ها دیده می‌شود.

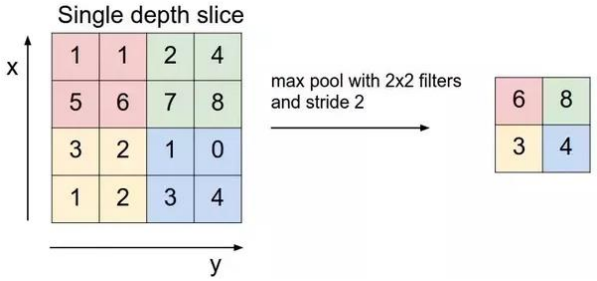
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

شکل ۷ چند نمونه از فیلترهای پرکاربرد و نتیجه‌ی اعمال هریک از فیلترها بر روی تصویر ورودی (سطر اول)

لایه‌ی ادغام[8]: این لایه برای کاهش ابعاد تصویر در شبکه استفاده می‌شود. در این لایه هدف این است که با حفظ اطلاعات مهم، تعداد پارامترها، کاهش یابد. ادغام بیشینه[9] یکی از انواع ادغام است که در آن در هر بخش از ورودی تنها مقدار بیشینه نگه داشته شده و سایر مقادیر دور ریخته می‌شود. در شکل ۸ نمونه‌ای از ادغام بیشینه دیده می‌شود.

[8] Pooling
[9] Max-Pooling

h.veisi@ut.ac.ir



شکل ۸ نمونه‌ی ادغام بیشینه

h.veisi@ut.ac.ir