به نام خدا

دانشکده علوم و فنون نوین

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

نیم‌سال اول ۱۴۰۱-۱۴۰۰

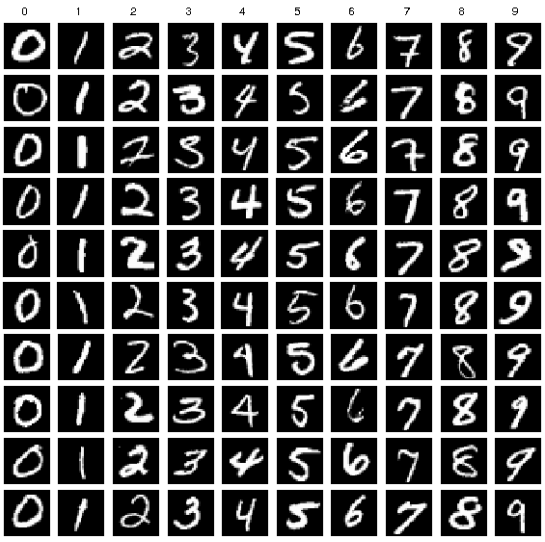
تمرین شماره ۲

تاریخ تحویل: ۱۴۰۰/۱۰/۰۳

---

۱. (۴۰٪) [**پیاده‌سازی: پرسپترون چندلایه برای تشخیص یک دسته**] در این تمرین با استفاده از یک شبکه‌ی پرسپترون چندلایه قرار است مدلی را آموزش دهیم که تصویر اعداد دست‌نویس مربوط به اعداد ۰ تا ۹ از مجموعه داده MNIST را تشخیص دهد. این مجموعه داده از تصاویر مربعی با ابعاد ۲۸*۲۸ پیکسل تشکیل شده که هرکدام حاوی یکی از ارقام صفر تا ۹ به صورت دست‌نویس است. در این مجموعه داده از هر رقم تعداد ۶ هزار تصویر در بخش آموزش (train) و هزار تصویر در بخش آزمایش (test) موجود است. نمونه‌هایی از این تصاویر در شکل زیر آورده شده است. مجموعه‌ی کامل این داده‌ها و توضیحات بیشتر در این مورد در لینک زیر موجود است.

http://yann.lecun.com/exdb/mnist



شکل ۱ نمونه تصاویر مجموعه دادگان MNIST

۱-۱ [۳۰٪] [**آموزش auto-encoder**] در این بخش به کمک شبکه‌ی پرسپترون چندلایه یک شبکه auto-encoder (مشابه با مثال فشرده‌سازی تصویر) برای همه اعداد در مجموعه داده‌ی MNIST آموزش می‌دهیم. برای این منظور لازم است الگوریتم پرسپترون چندلایه را به صورت کلی (با امکان تغییر تعداد لایه‌های مخفی، تعداد نرون‌های هر لایه، تنظیم تابع فعال‌ساز و...) پیاده‌سازی کنید. **توجه کنید که**

h.veisi@ut.ac.ir

به نام خدا

دانشکده علوم و فنون نوین

مبانی محاسبات (رایانش) نرم (۰۱-۰۳۹-۰۵-۸۳)
نیم‌سال اول ۱۴۰۱-۱۴۰۰

تمرین شماره ۲

تاریخ تحویل: ۱۴۰۰/۱۰/۰۳

تمامی مراحل مربوط به این الگوریتم باید توسط دانشجو پیاده‌سازی شود و استفاده از توابع آماده مجاز نیست.

با داشتن الگوریتم پرسپترون چندلایه می‌توان از شبکه‌ای با ۷۸۴ نرون ورودی (پیکسل‌های یک تصویر) و لایه‌ی میانی با تعداد نرون کمتر (مثلاً ۶۴) و لایه‌ای خروجی با ۷۸۴ نرون (تصویر بازسازی شده) به عنوان auto-encoder استفاده کرد. با آموزش چنین شبکه‌ای می‌توانیم تصویر ورودی را در لایه‌ی وسط شبکه encode کنیم و سپس مجدداً با بخش دوم شبکه، آن را decode کرده و تصویر اصلی را بازسازی کنیم. از Mean Squared Error به عنوان خطای بازشناسی استفاده کنید.

یک شبکه‌ی پرسپترون سه لایه با تعداد ۷۸۴، ۶۴ و ۷۸۴ نرون در هر لایه و با استفاده از تابع فعال‌ساز sigmoid ایجاد کنید و یادگیری آن را با نرخ یک هزارم انجام دهید (بدون پس‌انتشار با گشتاور و بدون به‌روز کردن دسته‌ای). در هریک از بخش‌های **الف** و **ب** تغییرات گفته شده را بر روی شبکه با همین پارامترها اعمال کنید.

نکته ۱: در این بخش از همه داده‌ها استفاده کنید و قسمتی از این داده‌های آموزش را به عنوان داده‌های validation جدا کرده و میزان خطای آن را معیاری برای زمان توقف الگوریتم و ارزیابی مدل قرار دهید. از هیچ‌یک از داده‌های آزمایش در آموزش استفاده‌ای نمی‌کنیم و آن‌ها را برای ارزیابی نهایی نگه می‌داریم.

نکته ۲: برای هریک از اجراهای خواسته شده نموداری از روند تغییرات خطای MSE شبکه بر روی داده‌های train و validation رسم کنید (این دو نمودار را روی یکدیگر قرار دهید). در گزارش خود با تحلیل این نمودارها برای هر بخش، تأثیر موارد خواسته شده بر روی «سرعت همگرایی بر اساس تعداد epoch» و «توان شبکه در یادگیری الگو» را بررسی کرده و علت آن را بیان کنید.

نکته ۳: همچنین یک تصویر نمونه از مجموعه دادگان (بخش validation) را در نظر بگیرید و برای هر بخش، خروجی بهترین شبکه (تصویر بازسازی شده) را با تصویر اصلی مقایسه کنید (مانند شکل ۲).

h.veisi@ut.ac.ir

به نام خدا

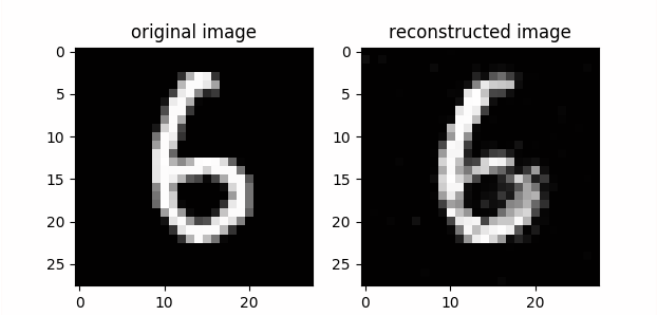
دانشکده علوم و فنون نوین

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

نیم‌سال اول ۱۴۰۱-۱۴۰۰

تمرین شماره ۲

تاریخ تحویل: ۱۴۰۰/۱۰/۰۳



شکل ۲ مقایسه‌ی تصویر اصلی و تصویر بازسازی شده توسط auto-encoder

الف) تعداد نرون‌های لایه‌ی مخفی را هریک از مقادیر ۶۴ و ۱۲۸ قرار دهید و برای هر حالت، بعد از آموزش، کارایی شبکه را بر روی داده آزمایش گزارش کنید. برای محاسبه کارایی، مقدار MSE بین تصویر اصلی آزمون و تصویر بازسازی شده در Decoder را حساب کرده و میانگین آن را گزارش کنید

ب) تعداد لایه‌های مخفی را از یک به ۳ و ۵ به صورت زیر تغییر دهید و مقدار میانگین MSE را روی داده آزمون در هر حالت گزارش کنید.

- تعداد نرون‌ها برای سه لایه‌ی مخفی به ترتیب ۲۵۶ – ۶۴ – ۲۵۶
- تعداد نرون‌ها برای پنج لایه‌ی مخفی به ترتیب ۲۵۶ – ۱۲۸ – ۶۴ – ۱۲۸ – ۲۵۶

پ) برای بهترین شبکه در بخش‌های قبل، تابع فعالسازی را به ReLU تغییر داده و کارایی آن را گزارش کنید.

ت) برای بهترین شبکه حاصل در بخش‌های قبلی (الف تا پ)، ضریب یادگیری را مقادیر یک میلیونم و یک صدم قرار دهید و نتیجه را گزارش دهید.

ث) برای بهترین شبکه حاصل در بخش‌های قبلی (الف تا پ)، از به‌روز کردن دسته‌ای، با اندازه‌های دسته ۸ و ۱۶ استفاده کنید و مقایسه کارایی آن‌ها با حالتی که از به‌روز کردن دسته‌ای استفاده نمی‌کنید، بیان کنید.

h.veisi@ut.ac.ir

ج) برای بهترین شبکه حاصل در بخش‌های قبلی، تعداد ۵ تصویر (از داده‌های آزمایشی) با بیشترین میزان خطای بازسازی را رسم کنید (مانند شکل ۲ آن‌ها را با تصویر اصلیشان مقایسه کنید). همین کار را برای ۵ تصویر با کمترین خطای بازسازی انجام دهید.

در نهایت با جمع‌بندی موارد بررسی شده بهترین پارامترهای پیشنهادی خود را بیان کنید و نتیجه‌گیری کدام شبکه و پارامترها بهترین هستند (از این شبکه برای قسمت بعدی سوال استفاده کنید).

۱-۲ [۱۰٪] [**دسته‌بندی اعداد**] با استفاده بهترین auto-encoder آموزش داده شده در سوال ۱-۱، می‌خواهیم یک دسته‌بند مبتنی بر MLP برای دسته‌بندی اعداد آموزش دهیم. برای این کار، داده‌های اعداد را از Encoder عبور داده و خروجی آن را به عنوان بردار ویژگی در نظر می‌گیریم. حال یک شبکه MLP سه لایه می‌سازیم با تعداد نرون‌های ورودی برابر با خروجی Encoder و تعداد نرون‌های خروجی برابر با ۱۰ (هر عدد یک نرون) و تعداد نرون دلخواه. این شبکه را با داده آموزش در بخش دادگان، آموزش داده و با داده آزمون ارزیابی کنید.

الف) شبکه MLP را با حداقل دو تعداد مختلف نرون در لایه مخفی آموزش دهید و مقدار Accuracy را روی داده آزمون محاسبه کنید. مقادیر پارامترهای مورد استفاده دیگر مانند نرخ یادگیری را نیز گزارش کنید.

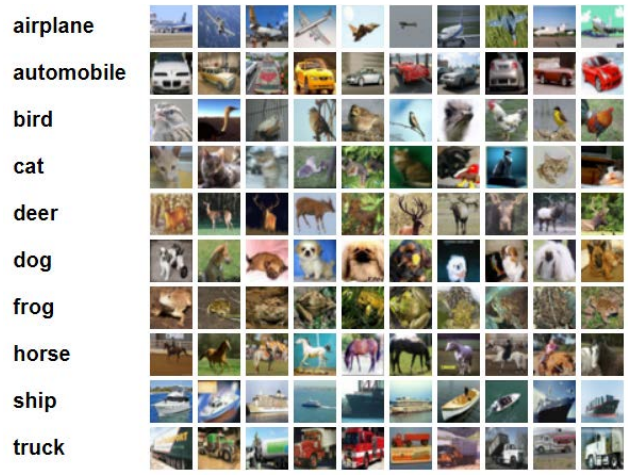
ب) بهترین نتایج خود را با بهترین نتایج گزارش شده در صفحه این دادگان مقایسه کنید، به ویژه با نتایج گزارش شده برای شبکه MLP. به نظر شما چطور می‌توان کارایی را بهتر کرد تا به بهترین نتایج گزارش شده در این وب سایت دست یافت؟

---

۲. (۷۰٪) [**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

h.veisi@ut.ac.ir

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 ۳ ملاحظه می‌شود.



شکل ۳ 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

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

h.veisi@ut.ac.ir

۴) لایه drop-out با احتمال ۳۰ درصد
۵) لایه خطی با تعداد ده (تعداد دسته‌ها) نرون خروجی

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 Adagrad

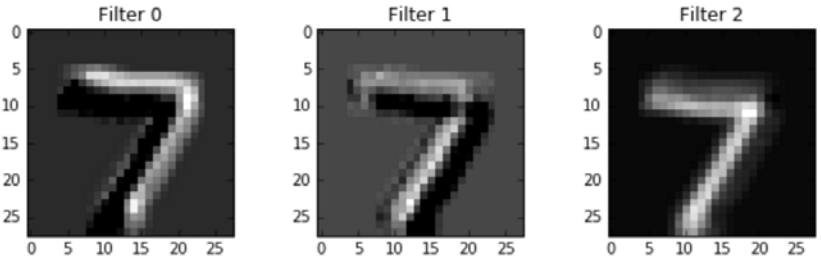
چ) اضافه کردن لایه‌ی Batch Normalization به عنوان لایه اول شبکه

[2] epoch

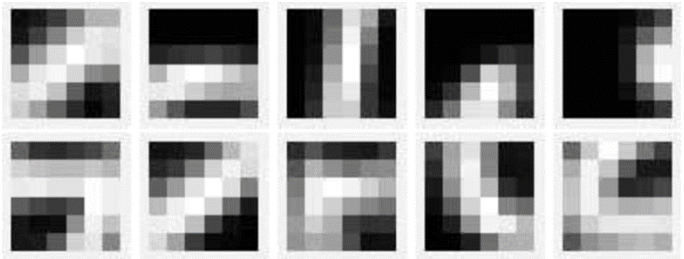
h.veisi@ut.ac.ir

ح) تغییر اندازه‌ی batch-size به ۴ و ۱۲۸

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 ۴). 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 ۵). 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 ۴ نمونه سه تصویر که توسط سه فیلتر مختلف در لایه‌های میانی شبکه‌ی عصبی تولید شده است



شکل ۵ 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

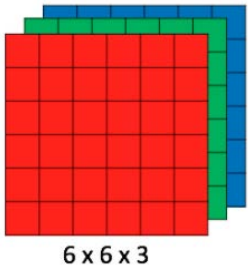
h.veisi@ut.ac.ir

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 ۶ 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.

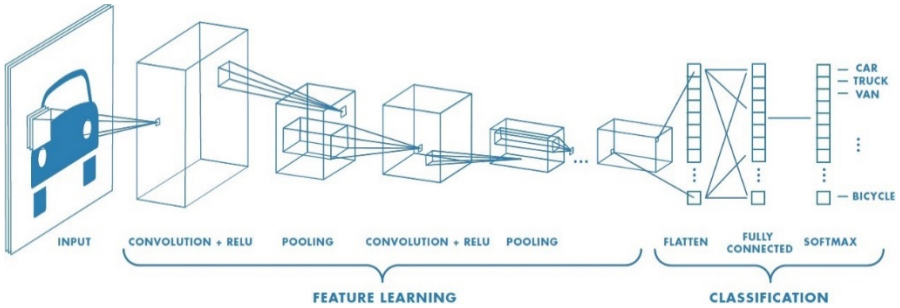


شکل ۶ نمایی از ماتریس سه‌بعدی یک تصویر رنگی با ابعاد ۶×۶

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 ۷ یک نمونه شبکه‌ی عصبی معمول که به وسیله‌ی آن دسته‌بندی تصاویر صورت می‌گیرد، ملاحظه می‌شود.

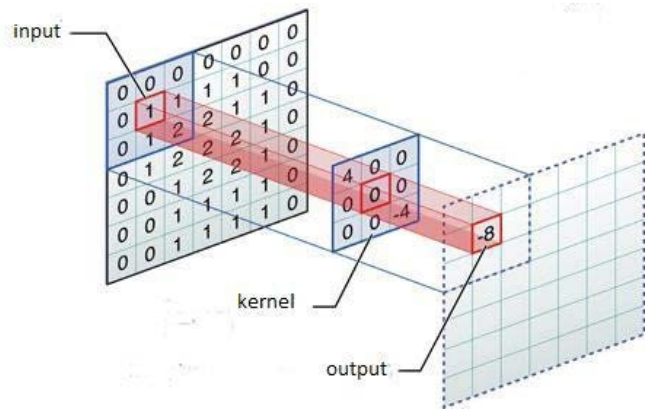
[7] Convolutional

h.veisi@ut.ac.ir



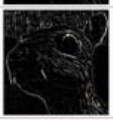
شکل ۷ نمونه معماری یک شبکه‌ی عصبی برای دسته‌بندی تصاویر

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 ۸ محاسبه می‌شود.



شکل ۸ 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 ۹ 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)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

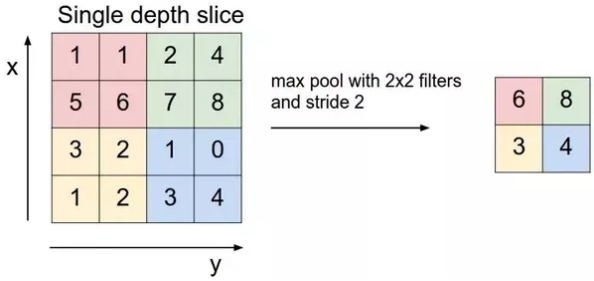
شکل ۹ چند نمونه از فیلترهای پرکاربرد و نتیجه‌ی اعمال هریک از فیلترها بر روی تصویر ورودی (سطر اول)

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 ۱۰ نمونه‌ای از ادغام بیشینه دیده می‌شود.

[8] Pooling
[9] Max-Pooling

h.veisi@ut.ac.ir

به نام خدا



شکل ۱۰ نمونه‌ی ادغام بیشینه

h.veisi@ut.ac.ir