



یادگیری ماشین

پاییز ۱۴۰۳

استاد: علی شریفی زارچی

گردآورندگان: امیرمهدی میقانی - علی رحیمی اکبر - پیام تائبی

زمان: ۳ ساعت

آزمون میان‌ترم دوم

۶ دی ۱۴۰۳

شماره دانشجویی:

نام و نام خانوادگی:

لطفا پاسخ هر سوال را زیر آن بنویسید. بالای هر برگه نام و شماره دانشجویی خود را حتما قید کنید. در پایان آزمون، برگه‌ی سوال‌های مختلف را از هم جدا کنید و هر برگه را در دسته‌ی مربوط به خود قرار دهید. این آزمون از ۱۱۰ نمره است و نمره کامل ۱۰۰ است. توجه کنید نمره‌ی بالای ۱۰۰ امتیازی نخواهد داشت.

درست نادرست (نمره ۲۵)

گزینه‌های درست را انتخاب کنید (میتوانید بیش از یک گزینه را انتخاب کنید). برای هر مورد مختصرا توضیح دهید چرا صحیح یا غلط است.

۱. کدام یک از موارد زیر درباره‌ی شبکه‌های عصبی کانولوشنی برای تحلیل تصاویر درست هستند؟

الف) فیلترها در لایه‌های اولیه تمایل به آشکار کردن لبه‌ها دارند.

ب) لایه‌های Pooling وضوح فضایی تصویر را کاهش می‌دهد.

ج) شبکه‌های عصبی کانولوشنی پارامترهای بیشتری نسبت به شبکه‌های کاملا متصل با همان تعداد لایه و تعداد یکسان نورون در هر لایه دارند.

د) یک CNN می‌تواند برای وظایف unsupervised learning آموزش داده شود، در حالی که شبکه عصبی عادی نمی‌تواند.

۲. شما شروع به آموزش شبکه عصبی خود می‌کنید، اما تابع هزینه کل (ترکیب cross entropy loss و loss regularization) از همان ابتدا تقریباً ثابت باقی مانده است. دلیل این مشکل چه می‌تواند باشد؟
(الف) نرخ یادگیری ممکن است خیلی پایین باشد.

(ب) Regularization Strength ممکن است خیلی زیاد باشد.

(ج) توزیع کلاس‌ها در داده ممکن است بسیار نامتعادل باشد.

(د) مقیاس مقداردهی اولیه وزن‌ها ممکن است به درستی تنظیم نشده باشد.

۳. کدام یک از موارد زیر درباره‌ی مدل‌های Auto Encoder درست هستند؟

(الف) در مدل‌های Sparse autoencoder کاهش تعداد node ها در Hidden layer مدل باعث ایجاد Information bottleneck می‌شود.

(ب) ایده کلی در مدل Sparse autoencoder این است که روند رمزگذاری و رمزگشایی تنها با فعال نگه داشتن تعداد کمی از نورون‌ها صورت بگیرد

(ج) یک راه برای ایجاد مدل undercomplete autoencoder این است که تعداد نورون‌های موجود در Hidden layers را کاهش دهیم.

(د) مدل‌های خودرمزگذار قادر به یادگیری منیفولدهای Manifold غیرخطی می‌باشند.

(ه) در مدل Denoising autoencoder ، تابع هزینه میان نسخه اصلی ورودی و بازسازی آن براساس نسخه نویزی آن، محاسبه می‌شود.

(و) از مدل Denoising autoencoder میتوان به عنوان ابزاری برای استخراج ویژگی استفاده کرد.

پاسخ سوال اول) گزینه‌های الف و ب

پاسخ سوال دوم) گزینه‌های الف و د

پاسخ سوال سوم)

الف نادرست و ب درست می‌باشد

مدل‌های خودرمزگذار پراکنده، بدون نیاز به regularization خاصی شامل گلوگاه اطلاعاتی می‌باشند. گزاره ب نیز تعریف این دسته از مدل‌ها می‌باشد.

ج درست و د درست می‌باشد

در گزاره اول، با کاهش تعداد نوروها می‌توان میزان اطلاعات در مدل خودرمزگذار را کاهش و در نتیجه آن را به مدل خودرمزگذار ناقص تبدیل کرد. در گزاره دوم، به صورت کلی مدل‌های خودرمزگذار را میتوان به عنوان یک ابزار کاهش ابعاد داده (dimensionality reduction) در نظر گرفت که برای این امر، مدل ابرصفحه‌هایی را برای تصویر کردن داده به ابعاد پایین‌تر را یاد می‌گیرد.

ه درست و درست می‌باشد

این مدل برای از بین بردن نیز در داده ورودی استفاده می‌شود، در نتیجه تابع هزینه زمان آموزش این مدل سعی می‌کند تفاوت میان نسخه اصلی داده ورودی و بازسازی داده براساس نسخه نویزی آن را کاهش دهد. خروجی بخش رمزگذار مدل (و ورودی بخش رمزگشا) ویژگی‌های استخراج‌شده از داده توسط مدل می‌باشند.

انتشار به عقب (۲۸ نمره)

۱- در این مسئله میخواهیم یک دسته بندی دوکلاسه انجام دهیم. اما دادگان ما بصورت نامتوازن هستند به صورتی که ۲۰۰۰ داده از کلاس اول و ۲۰۰ داده از کلاس دوم داریم.
(الف) یک راه حل برای حل عدم توازن دادگان ارائه دهید.

معماری شبکه استفاده شده بصورت زیر است (از چپ به راست):

$$z_1 = W_1 x^{(i)} + b_1 \quad a_1 = \text{ReLU}(z_1) \quad z_2 = W_2 a_1 + b_2 \quad (1)$$

$$\hat{y}^{(i)} = \sigma(z_2) \quad L^{(i)} = \alpha \cdot y^{(i)} \cdot \log(\hat{y}^{(i)}) + \beta \cdot (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \quad (2)$$

$$J = -\frac{1}{m} \sum_{i=1}^m L^i \quad (3)$$

ابعاد به شرح زیر است :

$$\hat{y}^{(i)} \in \mathbb{R}, y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^{D_x \times 1}, w_1 \in \mathbb{R}^{D_{w_1} \times D_x}, w_2 \in \mathbb{R}^{1 \times D_{w_2}} \quad (4)$$

توجه داشته باشد که m اندازه مجموعه داده ها است و دادگان بردارهایی با طول D_x هستند.
(ب) برای مقادیر زیر و به ازای ورودی داده شده خروجی شبکه را بدست آورید.

$$x = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, w_1 = \begin{bmatrix} 0.2 & 0.4 & -0.5 \\ -0.3 & 0.1 & 0.2 \end{bmatrix}, w_2 = [-0.3 \quad -0.2], b_1 = \begin{bmatrix} -0.4 \\ 0.2 \end{bmatrix}, b_2 = [0.1], \quad (5)$$

$$y = 1, \alpha = \beta = 0.5 \quad (6)$$

(ج) چرا α و β مفید هستند؟ یک جفت مقدار معقول برای α و β در این مسئله چیست؟ مقادیر مشخصی برای این وزن ها ارایه دهید.

(د) محاسبات مربوطه به الگوریتم انتشار به عقب را ابتدا بصورت پارامتری برای یک گام انجام داده و سپس وزن های داده شده را به ازای مقادیر داده شده اپدیت کنید. (همراه با نرخ یادگیری برابر با ۰.۱)

(ه) شما تصمیم میگیرید که به مدل L_2 Regularization اضافه کنید. تابع هزینه جدید خود را بنویسید و با استفاده از این تابع هزینه جدید، قانون بروزرسانی را برای w_1 بنویسید. اگر از L_1 Regularization استفاده میشد تابع هزینه چگونه میشد؟

پاسخ (الف) *Ausgmentation* های مختلف یکی از جواب ها است.
(ب)

$$z_1 = \begin{bmatrix} 0.2 & 0.4 & -0.5 \\ -0.3 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.4 \\ 0.2 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.1 \end{bmatrix} \quad (7)$$

$$a_1 = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix} \quad (8)$$

$$z_2 = \begin{bmatrix} -0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix} + \begin{bmatrix} 0.1 \end{bmatrix} = 0.08 \quad (9)$$

$$\hat{y}(i) = \frac{1}{1 + e^{-0.08}} \approx 0.52 \quad (10)$$

$$L(i) = 0.5(\log 0.52) \approx 0.142 \quad (11)$$

(ج) وزن دهی به اینکه هرکلاس چقدر در تابع ضرر مشارکت دارد می تواند به نزول گرادیان کمک کند، زیرا شبکه هنگام یادگیری از نمونه های کلاس کم تعدادتر، قدم های بزرگتری خواهد داشت. $\alpha = 0.1$ و $\beta = 1$ بطور تقریبی، نسبت باید جایی در حدود $\beta = 10 \cdot a$ باشد، اما نه خیلی بزرگ یا کوچک باشد.
(د)

$$\frac{\partial j}{\partial \hat{y}} = -\frac{1}{m} \sum_i \delta_1^{(i)} \quad (12)$$

که در آن :

$$\delta_1^{(i)} = a \cdot y^{(i)} / \hat{y}^{(i)} - \beta \cdot (1 - y^{(i)}) / (1 - \hat{y}^{(i)}) \quad (13)$$

$$\frac{\partial \hat{y}^{(i)}}{\partial z_2} = \delta_2 = \sigma(z_2)(1 - \sigma(z_2)) \quad (14)$$

$$\frac{\partial z_2}{\partial a_1} = \delta_2 = W_2 \quad (15)$$

$$\frac{\partial z_2}{\partial W_2} = \delta_2' = a_1^T \quad (16)$$

$$\frac{\partial z_1}{\partial W_1} = \delta_2 = \begin{cases} 0 & z_1 < 0 \\ 1 & z_1 \geq 0 \end{cases} \quad (17)$$

$$\frac{\partial z_{\mathfrak{I}}}{\partial W_{\mathfrak{I}}} = x^{(i)T} \quad (۱۸)$$

$$\frac{\partial J}{\partial W_{\mathfrak{I}}} = \delta_{\mathfrak{I}} = -\frac{1}{m} \sum_i \delta_{\mathfrak{I}}^{(i)} \cdot \delta_{\mathfrak{I}} \cdot (\delta_{\mathfrak{I}} \odot \delta_{\mathfrak{F}}) \cdot \delta_{\mathfrak{D}} \quad (۱۹)$$

$$\frac{\partial J}{\partial W_{\mathfrak{I}}} = \delta_{\mathfrak{I}} = -\frac{1}{m} \sum_i \delta_{\mathfrak{I}}^{(i)} \cdot \delta_{\mathfrak{I}} \delta'_{\mathfrak{I}} \quad (۲۰)$$

که در آن $\delta_{\mathfrak{I}}^{(i)}$ و $\delta_{\mathfrak{I}}$ اسکالر هستند و $\delta_{\mathfrak{I}}, \delta_{\mathfrak{F}} \in \mathbb{R}^{D_{w_{\mathfrak{I}}} \times 1}$ و $\delta_{\mathfrak{D}} \in \mathbb{R}^{1 \times D_x}$

$$W_{\mathfrak{I}}^{(\cdot)} = W_{\mathfrak{I}} - \eta \cdot \frac{\partial J}{\partial W_{\mathfrak{I}}} \quad (۲۱)$$

$$W_{\mathfrak{I}}^{(\cdot)} = W_{\mathfrak{I}} - \eta \cdot \frac{\partial J}{\partial W_{\mathfrak{I}}} \quad (۲۲)$$

(۵)

$$J_{\mathfrak{I}} = -\sum_i (\alpha \cdot (1 - y^{(i)}) \cdot \log(1 - \hat{y}^{(i)}) + \beta \cdot y^{(i)} \cdot \log(\hat{y}^{(i)})) + \frac{1}{\mathfrak{I}} \|W_{\mathfrak{I}}\|_{\mathfrak{I}} + \frac{1}{\mathfrak{I}} \|W_{\mathfrak{I}}\|_{\mathfrak{I}} \quad (۲۳)$$

$$J_{\mathfrak{I}} = -\sum_i (\alpha \cdot (1 - y^{(i)}) \cdot \log(1 - \hat{y}^{(i)}) + \beta \cdot y^{(i)} \cdot \log(\hat{y}^{(i)})) + \frac{1}{\mathfrak{I}} \|W_{\mathfrak{I}}\|_{\mathfrak{I}} + \frac{1}{\mathfrak{I}} \|W_{\mathfrak{I}}\|_{\mathfrak{I}} \quad (۲۴)$$

$$W_{\mathfrak{I}}^{(\cdot)} = W_{\mathfrak{I}} - \eta \cdot \left(\frac{\partial j}{\partial W_{\mathfrak{I}}} + \mathfrak{I} \cdot W_{\mathfrak{I}} \right) \quad (۲۵)$$

شبکه عصبی کانولوشنال (۲۱ نمره)

در این سوال، شما در حال تنظیم مقادیر هایپرپارامترهای لایه کانولوشنی اول یک شبکه عصبی کانولوشنی هستید. ورودی شبکه تصویری با ابعاد $3 \times 128 \times 128$ است. هدف شما محاسبه ابعاد خروجی (ابعاد فعال سازی) و تعداد پارامترهای ممکن برای تنظیمات مختلف این لایه است. فرمت ابعاد به صورت (H, W, C) است، که H ارتفاع، W عرض و C تعداد کانال ها را نشان می دهد. (توجه کنید که هنگام محاسبه ابعاد در صورت اعشاری شدن از قطع کردن باید استفاده شود.)

۱. اولین لایه کانولوشنی دارای stride برابر با ۱، size filter برابر با ۳، padding input برابر با ۰ و ۶۴ فیلتر است.

۲. اولین لایه کانولوشنی دارای stride برابر با ۱، size filter برابر با ۵، padding input برابر با ۲ و ۱۶ فیلتر است.

۳. اولین لایه کانولوشنی دارای stride برابر با ۲، size filter برابر با ۲، padding input برابر با ۰ و ۳۲ فیلتر است.

اکنون که شکل های خروجی و تعداد پارامترهای این تنظیمات را تعیین کردید، قصد دارید یک شبکه عصبی کانولوشنی عمیق تر ایجاد کنید. فرض کنید یک شبکه شامل سه ماژول یکسان طراحی کرده اید که هر کدام از یک لایه کانولوشنی، یک لایه Max-Pooling و یک لایه ReLU تشکیل شده اند. تمام لایه های Pooling دارای stride برابر با ۲ و عرض/ارتفاع ۲ هستند. به عنوان مثال، اگر لایه کانولوشنی دارای stride برابر با ۱، size filter برابر با ۱، padding input برابر با ۰ و ۸ فیلتر باشد، معماری هر ماژول به صورت زیر خواهد بود:

- کانولوشن 1×1 با stride برابر با ۱ و padding input برابر با ۰
- Max-Pool با ابعاد 2×2 و stride برابر با ۲
- ReLU

سه ماژول مشابه را متوالی کنار هم میگذاریم تا شبکه کامل را تشکیل دهند. با توجه به مقادیر زیر برای هایپرپارامترهای کانولوشن، ابعاد خروجی فعال‌سازی و تعداد پارامترهای شبکه کامل را محاسبه کنید.

۴. لایه‌های کانولوشنی دارای stride برابر با ۱، size filter برابر با ۳، padding input برابر با ۰ و ۶۴ فیلتر هستند.

۵. لایه‌های کانولوشنی دارای stride برابر با ۱، size filter برابر با ۵، padding input برابر با ۲ و ۱۶ فیلتر هستند.

۶. لایه‌های کانولوشنی دارای stride برابر با ۲، size filter برابر با ۲، padding input برابر با ۰ و ۳۲ فیلتر هستند.

۷. فرض کنید برای حل یک مسأله‌ی دسته‌بندی با ۱۰ کلاس، شما یک لایه‌ی کاملاً متصل (Fully-Connected) بعد از سه ماژول کانولوشن-پولینگ-رلو اضافه می‌کنید. ورودی این لایه، خروجی آخرین ماژول، به صورت flatten (کل داده را یک وکتور می‌کنیم)، است. از میان سه معماری قبلی (قسمت‌های ۴، ۵ و ۶)، کدام معماری تعداد پارامترهای کمتری برای لایه‌ی کاملاً متصل خواهد داشت؟ تعداد پارامترهای این لایه را نیز مشخص کنید.

پاسخ ۱:

- ابعاد فعال سازی: $۱۲۶ \times ۱۲۶ \times ۶۴$
- تعداد پارامترها: $۱۷۹۲ = ۶۴ \times (۳ \times ۳ \times ۳ + ۱)$

پاسخ ۲:

- ابعاد فعال سازی: $۱۲۸ \times ۱۲۸ \times ۱۶$
- تعداد پارامترها: $۱۲۱۶ = ۱۶ \times (۵ \times ۵ \times ۳ + ۱)$

پاسخ ۳:

- ابعاد فعال سازی: $۶۴ \times ۶۴ \times ۳۲$
- تعداد پارامترها: $۴۱۶ = ۳۲ \times (۲ \times ۲ \times ۳ + ۱)$

پاسخ ۴:

- ابعاد فعال سازی بعد از سه ماژول: $۱۴ \times ۱۴ \times ۶۴$
- تعداد پارامترها: $۷۵,۶۴۸ = ۶۴ \times ۲ \times (۳ \times ۳ \times ۶۴ + ۱) + ۶۴ \times (۳ \times ۳ \times ۳ + ۱)$

پاسخ ۵:

- ابعاد فعال سازی بعد از سه ماژول: $۱۶ \times ۱۶ \times ۱۶$
- تعداد پارامترها: $۱۴,۰۴۸ = ۱۶ \times ۲ \times (۵ \times ۵ \times ۱۶ + ۱) + ۱۶ \times (۵ \times ۵ \times ۳ + ۱)$

پاسخ ۶:

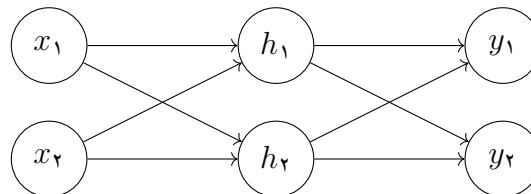
- ابعاد فعال سازی بعد از سه ماژول: $۲ \times ۲ \times ۳۲$
- تعداد پارامترها: $۸۶۷۲ = ۳۲ \times ۲ \times (۲ \times ۲ \times ۳۲ + ۱) + ۳۲ \times (۲ \times ۲ \times ۳ + ۱)$

پاسخ ۷:

- کمترین تعداد پارامترها مربوط به معماری قسمت ۶ است، با ابعاد فعال سازی $۲ \times ۲ \times ۳۲$.
- تعداد پارامترها: $۱۲۹۰ = ۱۰ \times (۲ \times ۲ \times ۳۲ + ۱)$

شبکه عصبی (۱۸ نمره)

در این سوال می‌خواهیم یک شبکه عصبی طراحی کنیم تا با دریافت ۴ مقدار، مرتب شده‌ی آن‌ها را خروجی بدهد. (آ) ابتدا با مسئله مرتب‌سازی دو عدد شروع می‌کنیم. شبکه‌ی عصبی زیر زوج مرتب (x_1, x_2) را به عنوان ورودی می‌گیرد و زوج مرتب (y_1, y_2) را به عنوان خروجی بر می‌گرداند به گونه‌ای که $y_1 = \max(x_1, x_2)$ و $y_2 = \min(x_1, x_2)$ می‌باشند. وزن‌ها و اریبی‌ها^۱ و توابع فعال‌سازی را در این شبکه مشخص کنید. (لازم نیست که توابع فعال‌سازی در هر لایه یکسان باشند)



(ب) اکنون با استفاده از شبکه ساخته شده در بخش آ، شبکه عصبی f را طراحی می‌کنیم تا با دریافت ورودی (x_1, x_2, x_3, x_4) مقادیر مرتب شده (y_1, y_2, y_3, y_4) را به عنوان خروجی بازگرداند. گراف این شبکه را رسم کنید. (برای انجام این کار می‌توانید شبکه پیاده‌سازی شده در بخش پیش را به عنوان یک ماژول در نظر بگیرید)

^۱ bias

(آ) می‌توانیم مقادیر را به صورت زیر بازنویسی کنیم:

$$y_1 = \frac{x_1 + x_2}{2} + \frac{|x_1 - x_2|}{2}$$

$$y_2 = \frac{x_1 + x_2}{2} - \frac{|x_1 - x_2|}{2}$$

یعنی داریم:

$$\rightarrow \begin{cases} x_1 \geq x_2 \rightarrow \begin{cases} y_1 = \frac{x_1 + x_2}{2} + \frac{x_1 - x_2}{2} = x_1 = \max(x_1, x_2) \\ y_2 = \frac{x_1 + x_2}{2} - \frac{x_1 - x_2}{2} = x_2 = \min(x_1, x_2) \end{cases} \\ x_1 < x_2 \rightarrow \begin{cases} y_1 = \frac{x_1 + x_2}{2} - \frac{x_1 - x_2}{2} = x_2 = \max(x_1, x_2) \\ y_2 = \frac{x_1 + x_2}{2} + \frac{x_1 - x_2}{2} = x_1 = \min(x_1, x_2) \end{cases} \end{cases}$$

در این صورت می‌توان گفت:

$$W^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad b^{(1)} = \begin{bmatrix} \cdot \\ \cdot \end{bmatrix} \quad \phi^{(1)}(x) = \begin{bmatrix} x_1 \\ |x_2| \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} \cdot/5 & \cdot/5 \\ \cdot/5 & -\cdot/5 \end{bmatrix} \quad b^{(2)} = \begin{bmatrix} \cdot \\ \cdot \end{bmatrix} \quad \phi^{(2)}(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(ب) با استفاده از مازول پیاده‌سازی شده در بخش پیش داریم:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \max(\max(x_1, x_2), \max(x_3, x_4)) \\ \max(\min(\max(x_1, x_2), \max(x_3, x_4)), \max(\min(x_1, x_2), \min(x_3, x_4))) \\ \min(\min(\max(x_1, x_2), \max(x_3, x_4)), \max(\min(x_1, x_2), \min(x_3, x_4))) \\ \min(\min(x_1, x_2), \min(x_3, x_4)) \end{bmatrix}$$

تعداد پارامتر (۱۸ نمره)

یک شبکه‌ی عصبی کانولوشنی تعریف شده توسط لایه‌های زیر در جدول ارائه شده است. ابعاد خروجی هر لایه و تعداد پارامترهای مربوط به آن را محاسبه کنید. می‌توانید شکل خروجی‌های فعال‌سازی را در قالب (H, W, C) بنویسید که در آن H ارتفاع، W عرض و C تعداد کانال‌ها را نشان می‌دهد. فرض کنید pad برابر ۱ و stride برابر ۱ است (مگر اینکه مشخص شده باشد).

- CONV_x-N یک لایه‌ی کانولوشن با N فیلتر با ابعاد $x \times x$ است.
- POOL- n یک لایه‌ی max-pooling با ابعاد $n \times n$ با گام یا stride برابر با n و بدون pad است.
- FLATTEN داده‌های ورودی را تخت می‌کند (کل داده‌ها را به صورت یک وکتور در می‌آورد).
- FC-N یک لایه‌ی کاملاً متصل با N نورون است.

Layer	Activation Volume Dimensions	Number of parameters
Input	$32 \times 32 \times 3$.
CONV3-8		
Leaky ReLU		
POOL-2		
BATCHNORM		
CONV3-16		
Leaky ReLU		
POOL-2		
FLATTEN		
FC-10		

Layer	Activation Volume Dimensions	Number of parameters
Input	$32 \times 32 \times 3$.
CONV3-8	$32 \times 32 \times 8$	$8 \times (3 \times 3 \times 3 + 1) = 224$
Leaky ReLU	$32 \times 32 \times 8$.
POOL-2	$16 \times 16 \times 8$.
BATCHNORM	$16 \times 16 \times 8$	2×8
CONV3-16	$16 \times 16 \times 16$	$16 \times (3 \times 3 \times 8 + 1) = 1168$
Leaky ReLU	$16 \times 16 \times 16$.
POOL-2	$8 \times 8 \times 16$.
FLATTEN	1024	.
FC-10	10	$(8 \times 8 \times 16 + 1) \times 10 = 10250$