



به نام خدا  
دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر



## درس شبکه‌های عصبی و یادگیری عمیق

### تمرین اول

نام دستیار طراح	عرفان باقری سولا	پرسش‌های ۱ و ۴
رایانامه	erfan.b.soula@ut.ac.ir	
نام دستیار طراح	فرید سیاهکلی	پرسش‌های ۲ و ۳
رایانامه	farbodsiahkali80@ut.ac.ir	
مهلت ارسال پاسخ	۱۴۰۱.۰۸.۱۶	

## فهرست

قوانین.....	۱
پرسش ۱. شبکه عصبی Mcculloch-Pitts.....	۱
۱-۱. نمایشگر 7-segment.....	۱
۲-۱. شبکه عصبی یک لایه.....	۲
۳-۱. شبکه عصبی دو لایه.....	۳
پرسش ۲ - آموزش شبکه‌های Adaline و Madaline.....	۵
۱-۲. Adaline.....	۵
۲-۲. Madaline.....	۵
پرسش ۳ - خوشه بندی با استفاده از Autoencoder.....	۷
۱-۳. پیاده‌سازی Deep Autoencoder برای کاهش ابعاد داده‌ها.....	۷
پرسش ۴. شبکه‌ی Multi-Layer Perceptron.....	۹
۱-۴. آشنایی و کار با مجموعه دادگان (پیش پردازش).....	۹
۲-۴. Teacher Network.....	۱۰
۳-۴. Student Networks.....	۱۱
۴-۴. Knowledge Distillation.....	۱۲

## شکل‌ها

- شکل ۱. نمایشگر هفت قسمتی ..... ۱
- شکل ۲. نحوه نمایش اعداد 6, 7, 8, 9 ..... ۲
- شکل ۳. ساختار شبکه عصبی یک لایه ..... ۲
- شکل ۴. ساختار شبکه عصبی دو لایه ..... ۳
- شکل ۵. نحوه تولید مجموعه داده مصنوعی ..... ۶
- شکل ۶. مدل پیشنهادی مقاله ..... ۸
- شکل ۷. نمودار تعداد داده‌ها به ازای هر کلاس برای داده‌های آزمون ..... ۹
- شکل ۸. معماری شبکه Teacher ..... ۱۰
- شکل ۹. معماری شبکه Student ..... ۱۱

## جدول‌ها

جدول ۱- عملکرد شبکه به ازای هر ورودی ..... ۲

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام **REPORTS\_TEMPLATE.docx** قرار داده شده تهیه نمایید.
- پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... هم‌گروه باشید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است؛** بنابراین، لطفاً تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
- **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند؛** بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **کدها حتماً باید در قالب نوت‌بوک با پسوند ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتماً در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت‌کننده در آن، 100- لحاظ می‌شود.**
- تنها زبان برنامه نویسی مجاز **Python** است.
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست.**

- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر (به ازای هر روز یک درصد تجمعی) وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number]\_[Lastname]\_[StudentNumber]\_[Lastname]\_[StudentNumber].zip

(مثال: HW1\_Ahmadi\_810199101\_Bagheri\_810199102.zip)

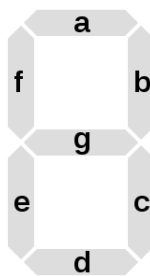
- برای گروه‌های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می‌شود هر دو نفر بارگذاری نمایند.

## پرسش ۱. شبکه عصبی Mcculloch-Pitts

در این پرسش ابتدا با نمایشگر هفت قسمتی آشنا خواهید شد و سپس دو شبکه عصبی برای طبقه‌بندی اعداد نمایش داده شده طراحی خواهید کرد. همچنین نقش نورون‌های لایه پنهان در شبکه‌های عصبی را بررسی خواهید کرد.

### ۱-۱. نمایشگر 7-segment

همان‌طور که در شکل ۱ مشاهده می‌شود، نمایشگر هفت قسمتی برای نشان دادن اعداد به صورت دیجیتالی استفاده می‌شود و دارای هفت LED (دیود نورانی) می‌باشد که روشن یا خاموش بودن این LED ها اعداد را به ما نشان می‌دهد.



شکل ۱. نمایشگر هفت قسمتی

مثلاً برای نمایش عدد ۱ کفایت که فقط حرف‌های b و c روشن بشوند.

فرض کنید یک نمایشگر 7-segment داریم که همواره یکی از اعداد 6, 7, 8, 9 را نمایش می‌دهد. در ادامه قصد داریم یک شبکه عصبی بسازیم که با داشتن وضعیت هر LED (روشن یا خاموش بودن) عدد در حال نمایش را تشخیص دهد. نحوه نمایش این اعداد را در شکل ۲ می‌توان مشاهده کرد.



شکل ۲. نحوه نمایش اعداد 6, 7, 8, 9

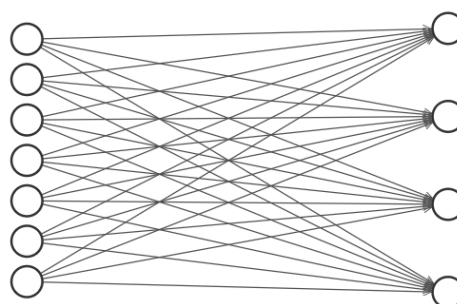
## ۲-۱. شبکه عصبی یک لایه

(۴۰ نمره)

در این قسمت یک شبکه عصبی یک لایه با ۴ نورون وجود دارد (شکل ۳) که خروجی هر نورون نشان دهنده مشاهده شدن یا نشدن یکی از اعداد است. این شبکه وضعیت هر LED را در ورودی دریافت می‌کند به طوری که اگر LED روشن باشد ورودی مربوطه ۱ و اگر خاموش باشد ورودی مربوطه ۰ است. خروجی این شبکه مطابق جدول ۱ است.

جدول ۱- عملکرد شبکه به ازای هر ورودی

a	b	c	d	e	f	g	$O_1$	$O_2$	$O_3$	$O_4$
1	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0	1	0	0
1	1	1	1	1	1	1	0	0	1	0
1	1	0	1	1	1	1	0	0	0	1
Any other combination							0	0	0	0



Input Layer  $\in \mathbb{R}^7$

Output Layer  $\in \mathbb{R}^4$

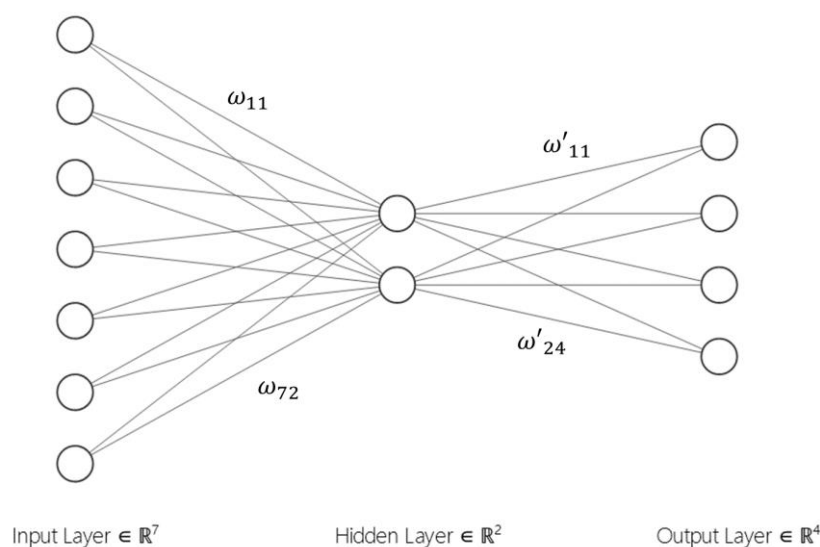
شکل ۳. ساختار شبکه عصبی یک لایه



حال با در نظر گرفتن  $w_{ij}$  به عنوان وزن شاخه‌ی بین ورودی  $i$  و نورون  $j$  و همچنین  $\theta_j$  به عنوان آستانه<sup>۱</sup> فعال‌سازی نورون  $j$ ، پارامترهای این شبکه را طوری تعیین کنید که خروجی عملکرد مورد نظر را داشته باشد. (توجه داشته باشید که نورون‌ها یک پارامتر threshold نیز دارند که در نمودار مشخص نشده است.)

### ۳-۱. شبکه عصبی دو لایه

در این قسمت یک شبکه عصبی با یک لایه پنهان وجود دارد (شکل ۴). لایه پنهان از دو نورون تشکیل شده که هر کدام وظیفه استخراج یک ویژگی یا الگو از ورودی را دارند. همچنین تضمین می‌شود که ورودی همواره یکی از اعداد 6, 7, 8, 9 می‌باشد و هیچ ترکیب دیگری در ورودی داده نمی‌شود. همانند بخش قبل وزن‌های بین لایه ورودی و لایه پنهان با  $w_{ij}$  و وزن‌های بین لایه پنهان و لایه خروجی با  $w'_{ij}$  نشان داده می‌شوند. همچنین آستانه فعال‌سازی نورون‌های لایه پنهان با  $\theta_j$  و آستانه فعال‌سازی نورون‌های لایه خروجی با  $\theta'_j$  نشان داده می‌شوند.



شکل ۴. ساختار شبکه عصبی دو لایه

**الف)** وزن‌های این شبکه را به گونه‌ای تعیین که خروجی مورد انتظار را به ازای هر ورودی ایجاد کند. (۴۰ نمره)

<sup>1</sup> threshold

ب) توضیح دهید که هر کدام از نوروں های لایه پنهان چه الگویی را تشخیص می دهند. (۱۵ نمره)

ج) تعداد پارامترهای شبکه یک لایه و دو لایه را محاسبه و مقایسه کنید. (۵ نمره)

## پرسش ۲ – آموزش شبکه‌های Adaline و Madaline

در این پرسش به بررسی دو روش Adaline و Madaline پرداخته خواهد شد.

### ۲-۱. Adaline

در این بخش با استفاده از روش Adaline یک شبکه عصبی آموزش داده خواهد شد که در مجموعه داده [iris](#) (که از ۳ نوع عنبیه مختلف Setosa، Versicolour و Virginica تشکیل شده)، نوع Setosa را از سایر دسته‌ها تشخیص دهد.

الف) ابتدا نمودار پراکندگی داده‌ها را در دو بعد رسم کنید (برای سادگی از دو ویژگی اول یعنی Sepal-width و Sepal-length استفاده شود)، سپس یک شبکه Adaline روی این داده‌ها آموزش دهید. همچنین در هنگام آموزش، نمودار تغییرات خطا، یعنی  $\frac{1}{2}(t - net)^2$  را رسم نمایید. (۲۰ نمره)

ب) حال این کار را برای عنبیه Versicolour انجام دهید (بدین معنی که مجموعه داده را به دو بخش Versicolour و Non-Versicolour تقسیم کرده و آموزش بر روی این داده‌ها انجام دهید). سپس دلیل خوب یا بد جدا شدن داده را نسبت به بخش الف توضیح دهید. (۲۰ نمره)

### ۲-۲. Madaline

در این بخش به پیاده سازی شبکه Madaline بر روی یک مجموعه داده مصنوعی پرداخته می‌شود.

الف) ابتدا به دلخواه یکی از الگوریتم‌های MRI و MRII را که در کتاب مرجع موجود است، انتخاب کرده و توضیح مختصری بدهید. (۱۰ نمره)

ب) برای آموزش از مجموعه داده مصنوعی، مطابق با آنچه در شکل ۵ نشان داده شده استفاده کنید. با استفاده از الگوریتمی که در بخش الف مطالعه نمودید، شبکه را آموزش دهید. سپس با تعداد نورون‌های متفاوت (یک بار ۳ نورون، یک بار ۵ نورون و یک بار ۸ نورون) نقاط را از هم جدا کنید. نهایتاً دقت جداسازی را در همه حالات نمایش دهید. (۴۰ نمره)

```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_moons

X_moons, Y_moons = make_moons(n_samples=500, noise=0.1, random_state=42)

X_train_moons, X_test_moons, Y_train_moons, Y_test_moons = train_test_split(X_moons,
Y_moons, test_size=0.5, random_state=3)

plt.scatter(X_moons[Y_moons == 0][:, 0], X_moons[Y_moons == 0][:, 1], color='red',
label='Class 0')
plt.scatter(X_moons[Y_moons == 1][:, 0], X_moons[Y_moons == 1][:, 1], color='blue',
label='Class 1')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.title("Moon-shaped Dataset")
plt.show()
```

شکل ۵. نحوه تولید مجموعه داده مصنوعی

ج) نمودارهای حاصل شده، دقت و تعداد ایپاک‌های هر سه حالت را با هم مقایسه و تحلیل کنید. (۱۰ نمره)

## پرسش ۳ – خوشه بندی با استفاده از Autoencoder

خوشه بندی داده‌ها یکی از مسائل مهم و پرکاربرد در زمینه‌ی یادگیری ماشین است و برای حل آن روش‌های متفاوتی وجود دارد. در این تمرین خوشه بندی با استفاده از AutoEncoder ها بررسی خواهد شد.

### ۳-۱. پیاده‌سازی Deep Autoencoder برای کاهش ابعاد داده‌ها

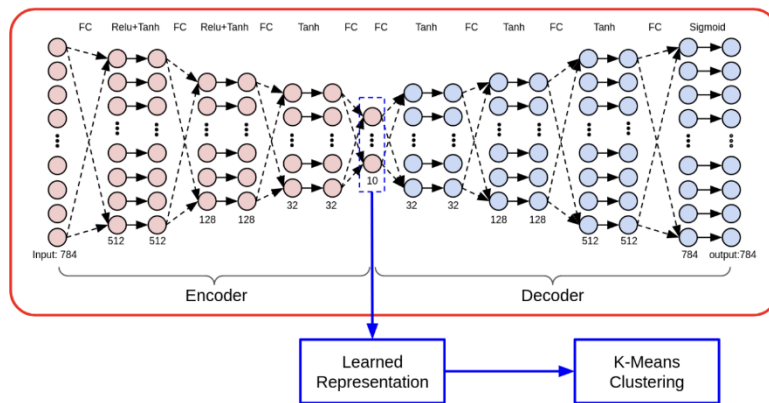
در این بخش با استفاده از مقاله " DAC: Deep Autoencoder-based Clustering, a General Deep Learning Framework of Representation Learning " یک Deep Autoencoder پیاده سازی خواهد شد. در این تمرین، از یک شبکه عمیق اتوانکودر (Deep Autoencoder) برای کاهش ابعاد داده‌ها و نمایش‌دهی بهتر به جهت انجام خوشه بندی با الگوریتم K-Means استفاده شده است.

خوشه‌بندی (Clustering) یک تکنیک یادگیری بدون نظارت است که هدف آن گروه‌بندی داده‌های مشابه صرفاً براساس ویژگی‌های موجود می‌باشد و برچسب‌های اولیه‌ای برای گروه‌ها وجود ندارد. هر گروه، شامل داده‌هایی است که ویژگی‌های شبیه‌تری نسبت به داده‌های باقی خوشه‌ها دارند.

الگوریتم K-Means یک الگوریتم خوشه‌بندی محبوب است که سعی می‌کند یک مجموعه از نقاط را به K خوشه تقسیم کند. این الگوریتم با انتخاب مراکز تصادفی برای هر خوشه کار کرده و هر داده را به نزدیک‌ترین مرکز اختصاص می‌دهد. برای مطالعه دقیق‌تر این مباحث می‌توانید از این [لینک](#) استفاده کنید.

برای پیاده سازی شبکه‌ی معرفی شده در این مقاله:

۱. ابتدا با استفاده از کتابخانه‌های مورد نیاز مانند TensorFlow و PyTorch یک شبکه عمیق اتوانکودر براساس مدل ارائه شده در مقاله (شکل ۶) ایجاد کنید. همچنین توجه کنید برای پیاده سازی باید از توابع فعالساز مختلفی مانند Sigmoid, Tanh و ReLU استفاده نمایید. (۱۰ نمره)



شکل ۶. مدل پیشنهادی مقاله

۲. تنها از مجموعه داده MNIST برای آموزش مدل استفاده کنید و اتوانکودر را با توجه به تابع خطای ارائه شده در مقاله (فرمول شماره ۶) آموزش دهید. (۴۰ نمره)

۴. پس از آموزش اتوانکودر، از لایه نهان حاصل از اتوانکودر به عنوان ورودی برای الگوریتم K-Means و برای انجام تسک clustering استفاده کرده و خروجی نهایی را ارزیابی نمایید. (۳۰ نمره)

۵. نتایج بدست آمده را با استفاده از معیارهای ارزیابی مقاله مانند Adjusted Rand Index (ARI) ارزیابی کرده و نتایجی همانند جداول مقاله گزارش دهید. (۲۰ نمره)

**(توجه)** این تمرین به عنوان مقدمه و ورود به مفاهیم شبکه‌های عصبی طراحی شده است. بنابراین، نیازی به پیچیدگی‌های بیشتر مانند تغییر معماری مدل یا تنظیم پارامترهای پیچیده نیست و صرفاً پیاده‌سازی درست مدل و الگوریتم آموزش مورد توجه است.

**(توجه)** می‌توانید از کتابخانه‌هایی مانند PyTorch یا TensorFlow برای پیاده‌سازی اتوانکودر و الگوریتم K-Means استفاده نمایید.

## پرسش ۴. شبکه‌ی Multi-Layer Perceptron

در این سوال با مجموعه داده MNIST آشنا خواهید شد و چند شبکه‌ی MLP را برای طبقه بندی این مجموعه داده آموزش خواهید داد. همچنین در ادامه با بحث Knowledge Distillation آشنا خواهید شد. برای آشنایی بیشتر با این موضوع مقاله ضمیمه شده را مطالعه کنید.

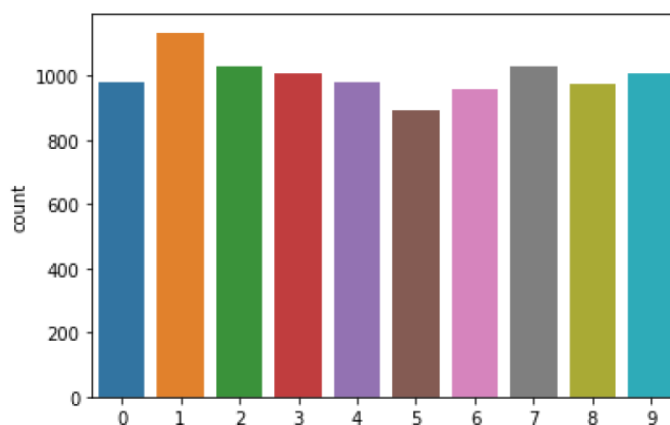
### ۴-۱. آشنایی و کار با مجموعه داده‌گان (پیش پردازش)

هدف از این قسمت، آشنایی و کار کردن با مجموعه داده مورد نظر است.

**الف)** ابتدا مجموعه داده‌گان MNIST را که مجموعه‌ای از ارقام دست‌نویس است فراخوانی کنید. برای این کار می‌توانید از توابع موجود در کتابخانه های Tensorflow/Keras و Pytorch استفاده نمایید. سپس تعداد و ابعاد داده‌های آموزش و آزمون را گزارش کنید. (۵ نمره)

**ب)** یک نمونه از هر کلاس را نمایش دهید. (۵ نمره)

**ج)** نمودار histogram مربوط به تعداد نمونه‌های هر کلاس را رسم کنید. این نمودار برای داده‌های آزمون رسم شده است (شکل ۷). همین نمودار را برای داده‌های آموزش رسم کنید. (۵ نمره)



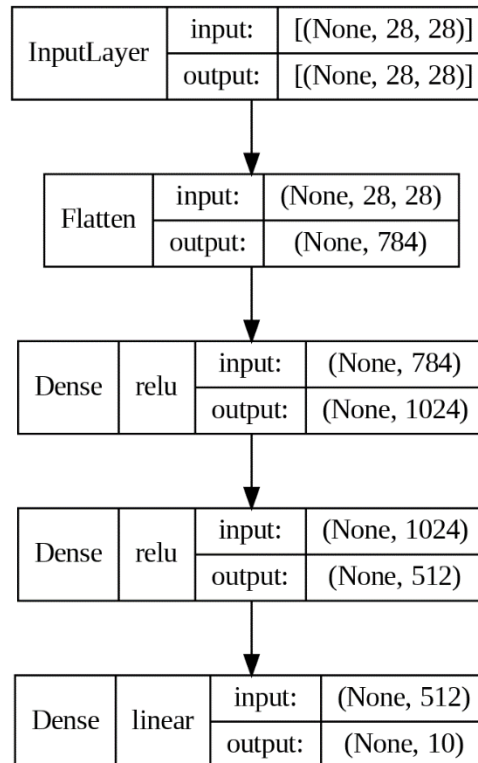
شکل ۷. نمودار تعداد داده ها به ازای هر کلاس برای داده های آزمون

**د)** با استفاده از min-max normalization داده‌ها را به بازه  $[0, 1]$  اسکیل کنید. (۵ نمره)

## Teacher Network ۲-۴

(۳۰ نمره)

ابتدا یک شبکه عصبی سه لایه را به ترتیب با ۱۰۲۴ و ۵۱۲ نورون در لایه پنهان به عنوان شبکه Teacher بسازید و در لایه های پنهان از تابع فعال سازی ReLU استفاده کنید (شکل ۸). دقت کنید که لایه آخر به صورت خطی است و تابع فعال ساز ندارد.



شکل ۸. معماری شبکه Teacher

حال این شبکه را با استفاده از تنظیمات زیر در ۲۰ epoch آموزش دهید، سپس نمودارهای accuracy و loss در طول آموزش را رسم کنید.

- Loss: Cross Entropy
- Optimizer: SGD
- Batch Size: 32
- Learning Rate: 0.01

در مسائل طبقه بندی، به خروجی هایی که از تابع فعال ساز softmax رد نشده اند logit گفته می شود. تابع هزینه Cross-Entropy در Pytorch به صورت پیش فرض logit دریافت می کند. اما در Tensorflow/Keras برای استفاده از logit به عنوان ورودی تابع هزینه Cross-Entropy باید هنگام فراخوانی کلاس مربوطه، پارامتر from\_logits=True را نیز به آن بدهید.

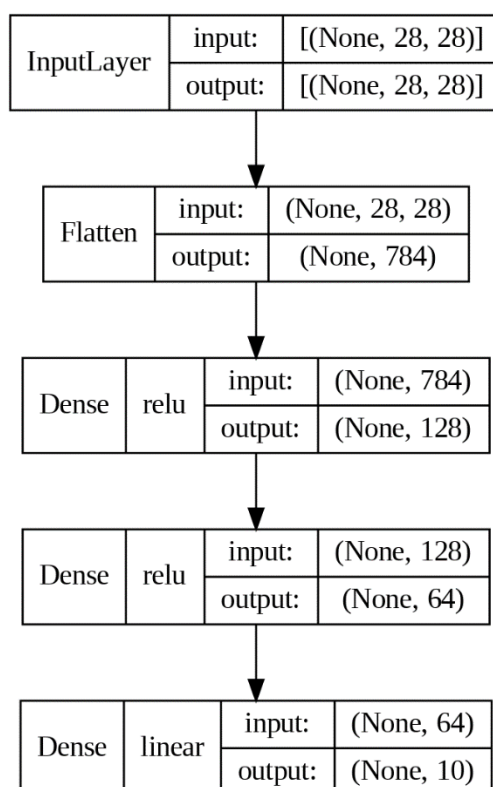


پس از پایان آموزش، accuracy شبکه روی داده‌های آزمون را گزارش کنید. سپس توضیح دهید که چگونه از روی logit های خروجی می‌توانیم کلاس پیش‌بینی شده توسط شبکه را پیدا کنیم. همچنین تعداد پیش‌بینی‌های غلط (misclassifications) شبکه روی داده‌های آزمون را نیز گزارش کنید.

## ۳-۴. Student Networks

(۲۰ نمره)

حال یک شبکه عصبی سه لایه را به ترتیب با ۱۲۸ و ۶۴ نورون در لایه پنهان به عنوان شبکه Student بسازید و در لایه های پنهان از تابع فعال سازی ReLU استفاده کنید (شکل ۹). دقت کنید که لایه آخر به صورت خطی است و تابع فعال ساز ندارد.



شکل ۹. معماری شبکه Student

این شبکه را با استفاده از تنظیمات زیر در ۱۰ epoch آموزش دهید. پس از پایان آموزش، تعداد پیش‌بینی‌های غلط (misclassifications) شبکه روی داده‌های آزمون را گزارش کنید.

- Loss: Cross Entropy

- Optimizer: SGD
- Batch Size: 32
- Learning Rate: 0.01

## ۴-۴ Knowledge Distillation

(۳۰ نمره)

در دنیای یادگیری ماشین، به فرایند انتقال دانش از یک مدل نسبتاً بزرگ به یک مدل کوچک Knowledge Distillation گفته می‌شود که یکی از روش‌های مرسوم برای انتقال اطلاعات بین مدل‌ها است و در چند سال گذشته بسیار مورد بررسی قرار گرفته است.

مدل‌های بزرگ مثل شبکه‌های عصبی عمیق یا مدل‌های Ensemble ظرفیت یادگیری بیشتری در مقایسه با مدل‌های کوچک‌تر دارند اما از لحاظ محاسباتی سنگین‌تر هستند. Knowledge Distillation این دانش را از مدل‌های پیچیده به مدل‌های ساده‌تر و کوچک‌تر منتقل می‌کند و تلاش می‌کند که این کار را با کمترین میزان هدر رفت دانش انجام دهد. در این فرایند مدل‌های بزرگ‌تر Teacher و مدل‌های کوچک‌تر Student نام دارند. یک راه برای انتقال دادن Generalization از مدل Teacher به مدل Student این است که از توزیع احتمال خروجی مدل Teacher که آن را Soft Target می‌نامند برای آموزش مدل Student استفاده کنیم.

برای این کار، ابتدا تمام داده‌های آموزش را یک بار به شبکه Teacher بدهید و logit‌های خروجی را آن را نگه دارید. سپس یک شبکه Student جدید همانند بخش قبل بسازید ولی این بار برای آموزش مدل Student به جای استفاده از لیبل‌های واقعی از logit‌های خروجی شبکه Teacher استفاده کنید به طوری که شبکه Student سعی کند خروجی‌های مدل Teacher را تقلید کند.

این شبکه را با استفاده از تنظیمات زیر در ۱۰ epoch آموزش دهید. پس از پایان آموزش، تعداد پیشبینی‌های غلط (misclassifications) شبکه روی داده‌های آزمون را گزارش کنید.

- Loss: MSE
- Optimizer: SGD
- Batch Size: 32
- Learning Rate: 0.01

\* دقت کنید که در اینجا باید از تابع هزینه Mean Squared Error استفاده کنید.

**الف)** تعداد پیشبینی‌های غلط (misclassifications) مدل Student بخش قبل روی داده‌های آزمون را با مدل Student فعلی مقایسه کرده و نتیجه را گزارش کنید.

ب) accuracy هر دو مدل Student روی داده های train در طول آموزش را در یک نمودار و در کنار هم رسم کرده و توضیح دهید که استفاده از Knowledge Distillation چه اثراتی داشته است.