



به نام خدا



دانشگاه تهران  
دانشکده ی مهندسی برق و کامپیوتر

هوش مصنوعی

گزارش پروژه سوم – **neural network**

نام و نام خانوادگی	پویا نقوی	روزبه بستان دوست
شماره ی دانشجویی	۸۱۰۱۹۴۴۱۷	۸۱۰۱۹۴۲۷۷
تاریخ ارسال گزارش	۲۸ فروردین ماه ۱۳۹۷	

## پاسخ سوالات بیان شده در صورت پروژه:

۱- در مرحله ی **feed forward** محتوای **node** های ورودی یک **node** در وزن یال های مربوط به آن ضرب شده و این حاصل ضرب ها با هم جمع می شود. در نهایت مقدار حاصل با یک **bias** جمع شده و مقدار **node** جدید را می سازد.

در مرحله ی **feed backward** میزان اختلاف خروجی شبکه با مقدار اصلی آن محاسبه می شود و طبق فرمول های مشتق زنجیره ای میزان خطای یال ها و بایاس مشخص شده و با توجه به نوع آن که می تواند **gradient descent** و یا **stochastic gradient descent** باشد به ترتیب در اولی بعد از اتمام کل داده ها وزن ها را آپدیت می کند و در دومی بعد از هر داده این کار را انجام می دهد.

۲- در روش **یادگیری با ناظر**، یک روش عمومی در یادگیری ماشین است که در آن به یک سیستم، مجموعه ای از جفت های ورودی-خروجی ارائه شده و سیستم تلاش می کند تا تابعی از ورودی به خروجی را فرا گیرد. یادگیری تحت نظارت نیازمند تعدادی داده ورودی به منظور آموزش سیستم است. با این حال رده ای از مسائل وجود دارند که خروجی مناسب که یک سیستم یادگیری تحت نظارت نیازمند آن است، برای آن ها موجود نیست. این نوع از مسائل چندان قابل جوابگویی با استفاده از یادگیری تحت نظارت نیستند. یادگیری تقویتی مدلی برای مسائلی از این قبیل فراهم می آورد.

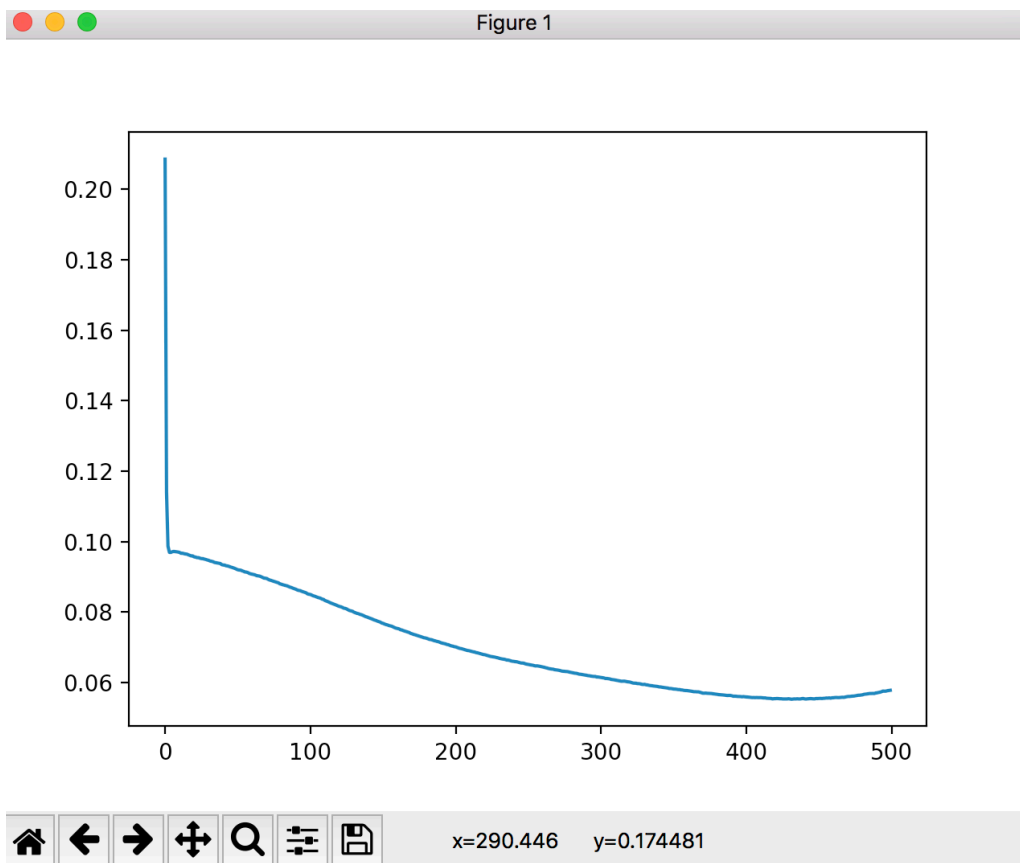
در **یادگیری تقویتی**، سیستم تلاش می‌کند تا تقابلات خود با یک محیط پویا را از طریق آزمون و خطا بهینه نماید. یادگیری تقویتی مسئله‌ای است که یک عامل که می‌بایست رفتار خود را از طریق تعاملات آزمون و خطا با یک محیط پویا فرا گیرد، با آن مواجه است. در یادگیری تقویتی هیچ نوع زوج ورودی- خروجی ارائه نمی‌شود. به جای آن، پس از اتخاذ یک عمل، حالت بعدی و پاداش بلافاصله به عامل ارائه می‌شود. هدف اولیه برنامه‌ریزی عامل‌ها با استفاده از تنبیه و تشویق است بدون آنکه ذکر از چگونگی انجام وظیفه آن‌ها شود.

در **یادگیری بدون ناظر** یا یادگیری خود سامانده، پارامترهای شبکه عصبی تنها توسط پاسخ سیستم اصلاح و تنظیم می‌شوند. به عبارتی تنها اطلاعات دریافتی از محیط به شبکه را بردارهای ورودی تشکیل می‌دهند.

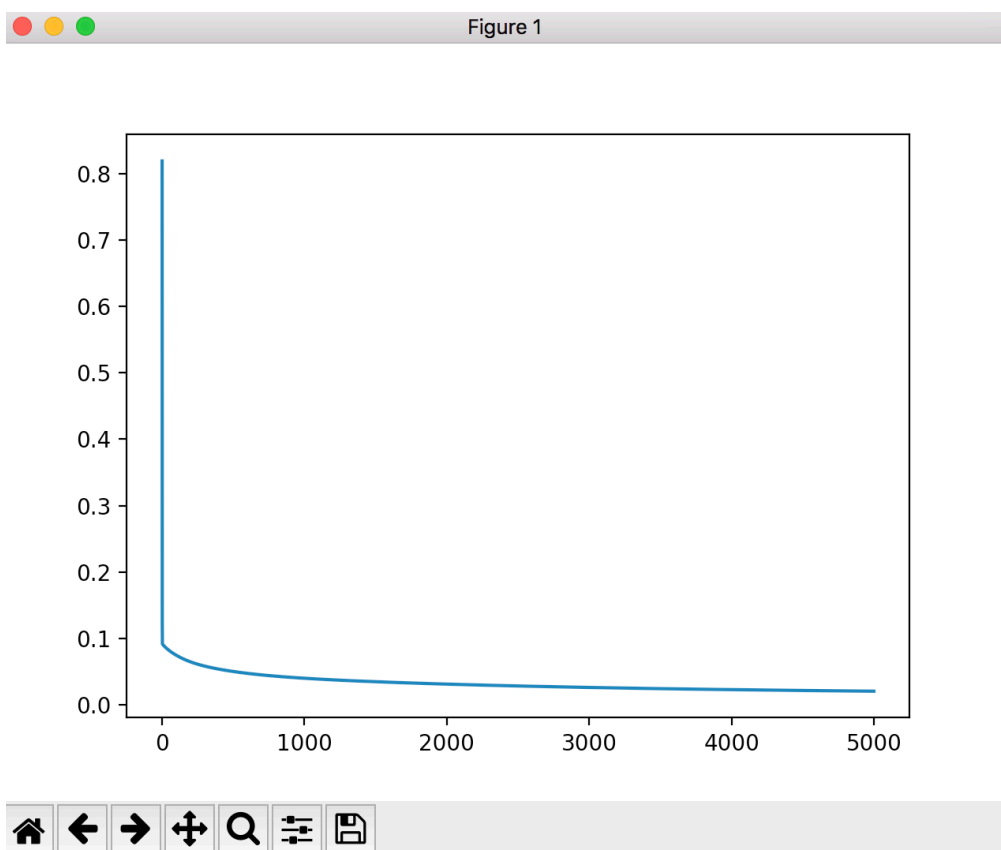
۳- بیش‌برازش (**Overfitting**) به پدیده‌ی نامطلوبی در آمار گفته می‌شود که در آن درجه آزادی مدل بسیار بیشتر از درجه‌ی آزادی واقعی انتخاب شده و در نتیجه اگرچه مدل روی داده استفاده شده برای یادگیری بسیار خوب نتیجه می‌دهد، اما بر روی داده جدید دارای خطای زیاد است. تنظیم کردن (**Regularization**) از راه‌های مقابله با این پدیده است که دو نوع آن **drop out** و **L<sub>2</sub>Norm** در سوال بعدی مفصل توضیح داده شده است.

۴- **Dropout** در لغت به معنی رها کردن و حذف تصادفی است و در شبکه‌های عصبی هم دقیقاً به همین منظور به کار می‌رود. به زبان خیلی ساده وقتی **dropout** استفاده می‌کنیم به این معناست که برخی از نورون‌ها در هنگام آموزش به صورت تصادفی نادیده بگیریم. به طور خاص‌تر به این معناست که با احتمال **p** یا **۱-p** این نورون‌ها را آپدیت کنیم و یا نکنیم. در شبکه‌های شلوغ که **fully connected** هستند، این **regularization** بیشتر کارایی دارد چرا که شبکه را خلوت می‌کند.

در **L<sub>2</sub>Norm** سعی بر پیدا کردن ضریبی است تا خروجی لایه‌ی **hidden** را رو به صفر متمایل کند و از افزایش بیش از اندازه و یا منفی شدن بیش از اندازه جلوگیری کند. در شبکه‌هایی که وابستگی خطی دارند، این حالت بیشتر کارایی دارد.



stochastic



gradient descent

۶- شبکه‌های عصبی بازگشتی (recurrent neural network)، شبکه‌هایی هستند که در ساختار آنها یال‌های بازگشت کننده وجود دارد. یعنی گراف آن‌ها جهت‌دار است. بر خلاف شبکه‌های عصبی رو به جلو، یال‌ها می‌توانند تشکیل دور بدهند. مزیت آن این است که **back propagation** در آن راحت‌تر می‌شود و عیب آن این است که هزینه‌ی تولید گراف افزایش پیدا می‌کند.

شبکه‌های عصبی پیچشی (convolutional neural network) رده‌ای از شبکه‌های عصبی عمیق هستند که معمولاً برای انجام تحلیل‌های تصویری یا گفتاری در یادگیری ماشین استفاده می‌شوند. شبکه‌های عصبی پیچشی به منظور کمینه کردن پیش‌پردازش‌ها از گونه‌ای از پرسپترون‌های چندلایه استفاده می‌کنند. ساختار شبکه‌های پیچشی از فرایندهای زیستی قشر بینایی گربه الهام گرفته شده است. این ساختار به گونه‌ای است که تک‌نورون‌ها تنها در یک ناحیه محدود به تحریک پاسخ می‌دهند که به آن ناحیه پذیرش گفته می‌شود. نواحی پذیرش نورون‌های مختلف به صورت جزئی با هم همپوشانی دارند به گونه‌ای که کل میدان دید را پوشش می‌دهند. شبکه‌های عصبی پیچشی نسبت به بقیه رویکردهای دسته‌بندی تصاویر به میزان کمتری از پیش‌پردازش استفاده می‌کنند. این امر به معنی آن است که شبکه معیارهایی را فرا می‌گیرد که در رویکردهای قبلی به صورت دستی فراگرفته می‌شدند. این استقلال از دانش پیشین و دستکاری‌های انسانی در شبکه‌های عصبی پیچشی یک مزیت اساسی است. مزیت آن این است که شبکه سبک‌تر می‌شود و عیب آن این است که تعدادی از وزن‌ها را در نظر نمی‌گیریم.

## ۱. چکیده

در این پروژه درباره یکی از کاربرد های شبکه عصبی و نحوه یادگیری آن اطلاعات کسب شد و توانستیم یکی از مدل های آن را پیاده سازی کنیم.

## ۲. ارائه‌ی روش

برای بهسازی الگوریتم از روش گرادیان کاهشی و گرادیان کاهشی تصادفی، برای تابع فعالسازی از روش های خطی و غیر خطی سیگموندی و برای Regularization از روش های Dropout و  $L_2$  Norm استفاده کردیم.

## ۳. ارائه‌ی نتایج

در سوال ۵ نتایج گفته شده نمایش داده شده است.

## ۴. تحلیل نتایج

با توجه به back propagation و تاثیر دادن خطا و آپدیت کردن یال‌ها، کم کم خطاها کاهش می‌یابد.

## ۵. جمع‌بندی و نتیجه‌گیری

برای یادگیری می‌توان از روش شبکه‌های عصبی استفاده کرد و با توجه به نتایج بدست آمده و نتایج اصلی وزن‌ها را آپدیت می‌کنیم و این کار را انقدر ادامه می‌دهیم تا جواب‌ها به اندازه‌ی کافی به جواب درست نزدیک شود.

```
..s/Assignment3
[ 4956 ]: 0.0219305557037565
[ 4957 ]: 0.021928468588797122
[ 4958 ]: 0.021926479328516348
[ 4959 ]: 0.021924322126461174
[ 4960 ]: 0.021922265006991078
[ 4961 ]: 0.02192030981022376
[ 4962 ]: 0.02191838184803839
[ 4963 ]: 0.021916380115358434
[ 4964 ]: 0.021914287561703938
[ 4965 ]: 0.021912201294711792
[ 4966 ]: 0.02191012584401285
[ 4967 ]: 0.021908023257070176
[ 4968 ]: 0.021905998934302996
[ 4969 ]: 0.021903950139622315
[ 4970 ]: 0.021901902179021814
[ 4971 ]: 0.021899907533360883
[ 4972 ]: 0.021897932247277985
[ 4973 ]: 0.021895957976606713
[ 4974 ]: 0.02189401525171463
[ 4975 ]: 0.021892046240534695
[ 4976 ]: 0.021890065967509094
[ 4977 ]: 0.021888097117154347
[ 4978 ]: 0.021886086206299945
[ 4979 ]: 0.021884072110801154
[ 4980 ]: 0.02188201006921172
[ 4981 ]: 0.021880026471278863
[ 4982 ]: 0.021878045964097988
[ 4983 ]: 0.021876134871043472
[ 4984 ]: 0.02187417031811777
[ 4985 ]: 0.021872212936853205
[ 4986 ]: 0.021870133540330565
[ 4987 ]: 0.02186817996565288
[ 4988 ]: 0.021866116996774865
[ 4989 ]: 0.02186408421212839
[ 4990 ]: 0.021862032509606517
[ 4991 ]: 0.021860049132286303
[ 4992 ]: 0.021858135890064224
[ 4993 ]: 0.021856121478527283
[ 4994 ]: 0.021854077860292094
[ 4995 ]: 0.021852115973472752
[ 4996 ]: 0.021850051407201695
[ 4997 ]: 0.021847919302042395
[ 4998 ]: 0.021845823766735557
[ 4999 ]: 0.021843850632801143
Training Acc= 91.5 %
New Data Acc= 84.63333333333334 %
(venv) ~ pooya at Pooya's MacBook Pro in ~/Desktop/University/Term 6(S 97)/Artificial_Intelligence/Projects/Assignment3 using
```

gradient descent

```
..s/Assignment3
[ 456 ]: 0.06517539591174487
[ 457 ]: 0.0653576826391674
[ 458 ]: 0.0654018044390011
[ 459 ]: 0.06545066952526149
[ 460 ]: 0.06555269330407532
[ 461 ]: 0.06560910069422551
[ 462 ]: 0.06578426574138865
[ 463 ]: 0.06579318444486651
[ 464 ]: 0.06588287233625503
[ 465 ]: 0.06600660558988979
[ 466 ]: 0.06608046846901391
[ 467 ]: 0.06617712468844386
[ 468 ]: 0.06626879610438699
[ 469 ]: 0.06630690155968386
[ 470 ]: 0.06645518524447933
[ 471 ]: 0.06654443982718332
[ 472 ]: 0.06659848247023431
[ 473 ]: 0.06663350337639742
[ 474 ]: 0.06664763010763328
[ 475 ]: 0.06677025258219646
[ 476 ]: 0.06684531503868447
[ 477 ]: 0.0669506110265434
[ 478 ]: 0.06692267195806577
[ 479 ]: 0.06699504954756483
[ 480 ]: 0.06720929357077296
[ 481 ]: 0.06724673454725502
[ 482 ]: 0.06726271245661383
[ 483 ]: 0.06739893396089801
[ 484 ]: 0.0674626350849549
[ 485 ]: 0.06758122029044795
[ 486 ]: 0.06766326800308485
[ 487 ]: 0.0677378075527765
[ 488 ]: 0.06781288148564033
[ 489 ]: 0.06791879746977462
[ 490 ]: 0.06791837997340938
[ 491 ]: 0.06804690272507954
[ 492 ]: 0.06805842448976412
[ 493 ]: 0.06820962324094414
[ 494 ]: 0.06832138951094072
[ 495 ]: 0.06845407283787334
[ 496 ]: 0.06851271607314041
[ 497 ]: 0.06855278882276779
[ 498 ]: 0.06855140310777647
[ 499 ]: 0.06857664474868555
Training Acc= 69.43333333333334 %
New Data Acc= 63.53333333333333 %
(venv) pooya at Pooya's MacBook Pro in ~/Desktop/University/Term 6(S 97)/Artificial_Intelligence/Projects/Assignment3 using
```

stochastic