پروژه دوم – رگولاریزاسیون

مدلهای یادگیری عمیق انعطاف پذیری و ظرفیت بسیار زیادی دارند که اگر مجموعه داده آموزش به اندازه کافی بزرگ نباشد، بیش برازش می تواند یک مشکل جدی باشد. مطمئناً چنین شبکه ای در مجموعه آموزش خوب عمل می کند، اما شبکه آموخته شده به نمونه های جدیدی که هرگز ندیده تعمیم نمی یابد!

بیان مساله

شما به تازگی به عنوان یک متخصص هوش مصنوعی توسط شرکت فوتبال فرانسه استخدام شده اید. آنها دوست دارند موقعیت هایی را پیشنهاد دهید که دروازه بان فرانسه باید توپ را لگد بزند تا بازیکنان تیم فرانسه بتوانند آن را با سر خود بزنند.

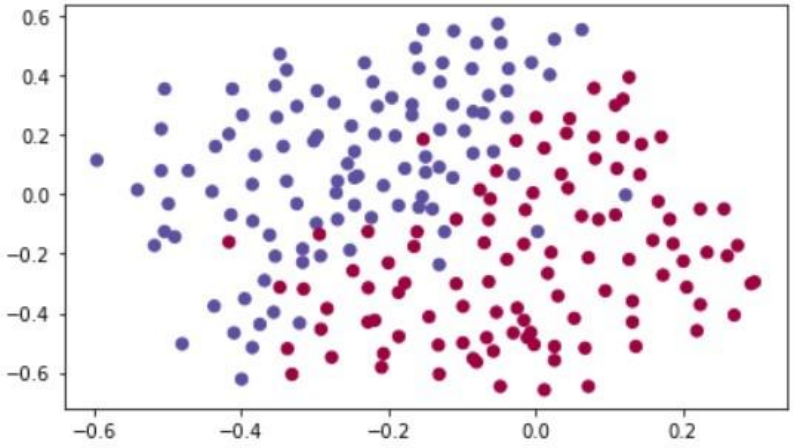
آنها یک مجموعه داده ۲ بعدی از ۱۰ بازی گذشته فرانسه به شما می دهند.

مرحله صفر: اضافه کردن کتابخانه های لازم

محتویات پوشه Requirements را در مسیر پروژه اضافه کنید.

مرحله یک: بارگذاری مجموعه داده ها

هر نقطه مربوط به موقعیتی در زمین فوتبال است که یک بازیکن فوتبال پس از شوت توپ از سمت دروازه بان، به توپ با سر خود ضربه زده است.



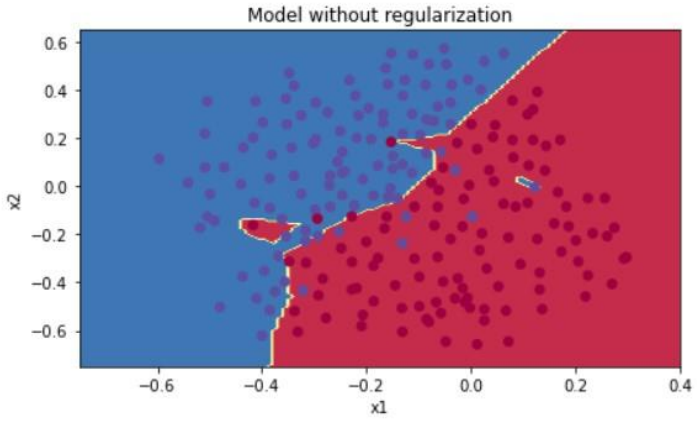
اگر نقطه آبی باشد، به این معناست که بازیکن فرانسوی موفق شد به توپ با سر خود ضربه بزند. اگر نقطه قرمز باشد، به این معنی است که بازیکن تیم دیگر با سر به توپ ضربه می زند.

هدف شما: برای یافتن موقعیت هایی در زمین که دروازه بان باید توپ را به آن سمت بزند، از یک مدل یادگیری عمیق استفاده کنید. این مجموعه داده دارای کمی نویز است ، اما به نظر می رسد یک خط مورب که نیمه بالایی سمت چپ (آبی) را از نیمه پایین سمت راست (قرمز) جدا می کند به خوبی کار می کند.

مرحله دو: مدل بدون رگولاریزاسیون

در ابتدا پیاده سازی مدل شبکه عصبی را بدون رگولاریزاسیون بررسی کنید و عملکرد آن را بر روی داده ها بررسی کنید.

آنچه مشاهده خواهید کرد نموداری به صورت زیر است که به وضوح بیش برازش شبکه را نشان میدهد.



برای برطرف کردن بیش برازش مدل از روش های رگولاریزاسیون استفاده می شود.

مرحله سه: رگولاریزاسیون با استفاده از L2

یکی از روش های استاندارد رگولاریزاسیون روش L2 است که در آن تابع هزینه از رابطه (۱) به شکل رابطه (۲) پیاده سازی می شود:

$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\left(a^{[L](i)}\right) + (1-y^{(i)})\log\left(1-a^{[L](i)}\right)\right) \tag{1}$$

$$J_{regularized} = \underbrace{-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\left(a^{[L](i)}\right) + (1-y^{(i)})\log\left(1-a^{[L](i)}\right)\right)}_{\text{cross-entropy cost}} + \underbrace{\frac{1}{m}\frac{\lambda}{2}\sum_{l}\sum_{k}\sum_{j}W_{k,j}^{[l]2}}_{\text{L2 regularization cost}} \tag{2}$$

تمرین اول: تابع compute_cost_with_regularization() را برای محاسبه رگولاریزاسیون L2 پیاده سازی کنید.

مطمئناً ، چون هزینه را تغییر داده اید، مجبورید Backpropagation را نیز تغییر دهید! تمام شیب ها باید با توجه به این هزینه جدید محاسبه شوند.

تمرین دوم: تابع backward_propagation_with_regularization را پیاده سازی کنید.

در نهایت مدل سازی را انجام داده و عملکرد آن را بررسی کنید.

مرحله چهار: Dropout

سرانجام، Dropout یک تکنیک رگولاریزاسیون است که به طور گسترده ای مورد استفاده قرار می گیرد و مخصوص یادگیری عمیق است. در هر تکرار به طور تصادفی برخی از سلول های عصبی را خاموش می کند.

هنگامی که برخی از سلول های عصبی را خاموش می کنید، در واقع مدل خود را تغییر می دهید. ایده پشت Dropout این است که در هر تکرار، شما مدل متفاوتی را آموزش می دهید که فقط از زیرمجموعه ای از نورون های شما استفاده می کند.

تمرین سوم: تکنیک Dropout را برای حرکت رو به جلوی شبکه (Forward Propagation) در مدل سه لایه پیاده سازی کنید. به خاطر داشته باشید که این روش برای لایه خروجی به کار نمی رود.

طبق آنچه که در ویدیوهای درس گفته شد، آرایه $d^{[1]}$ را با اندازه برابر با آرایه $a^{[1]}$ قرار دهید. هر عنصر از آرایه $d^{[1]}$ با احتمال keep_prob یک (نورون فعال) و در غیر اینصورت صفر (نورون خاموش) خواهد بود، که در شبه کد زیر نشان داده شده است:

```
for i,v in enumerate(x):
    if v < keep_prob:
        x[i] = 1
    else: # v >= keep_prob
        x[i] = 0
```

تمرین چهارم: تکنیک Dropout را برای حرکت رو به عقب شبکه (Back Propagation) در مدل سه لایه پیاده سازی کنید.

قبلاً با استفاده از ماسک d[l]، برخی از سلولهای عصبی را هنگام انتشار رو به جلو خاموش کرده بودید. در Backpropagation، با استفاده مجدد از همان ماسک روی dA1 باید همان نورون ها را ببندید. در هنگام حرکت رو به جلو ، شما A1 را به Keep_prob تقسیم کرده اید. بنابراین، در Backpropagation ، باید dA1 را به keep_prob تقسیم کنید.

در نهایت عملکرد هر سه مدل را با هم مقایسه کنید و بررسی کنید کدامیک از تکنیک های رگولاریزاسیون بهتر عمل کرده است.