استفاده از شبکه عصبی مصنوعی (یادگیری ماشین) در پی‌اچ‌پی

علا عالم فلکی[1]

چکیده:

تکنیک شبکه عصبی مصنوعی[2] یک مدل محاسباتی با توجه به ساختار و توابع شبکه عصبی زیستی است. یک شبکه عصبی با توجه جریان ورودی اطلاعات پیشرفت می‌کند، بنابراین جریان اطلاعات ورودی به شبکه، ساختار شبکه عصبی را تحت تاثیر قرار می‌دهد. (فرآیند یادگیری)

در زبان‌های برنامه‌نویسی این تکنیک می‌تواند جایگزین مناسبی برای برنامه‌نویسی قانون‌محور[3] باشد. در سناریویی نظیر بررسی اسپم بودن یک نظر، استفاده از برنامه‌نویسی قانون‌محور نیاز به تعریف قوانینی دارد که امکان نادرست بودن آنها با توجه به شرایط زیاد است. اما با استفاده از تکنیک شبکه‌های عصبی مصنوعی می‌توان با پردازش ورودی در شرایط متفاوت، تصمیم درست گرفت.

در این مقاله راجع‌به مقدمه‌ای از شبکه‌های عصبی مصنوعی و شیوه پیاده‌سازی آنها در زبان پی‌اچ‌پی بحث خواهیم کرد.

کلمات کلیدی: شبکه عصبی مصنوعی، هوش مصنوعی، یادگیری ماشین، پی‌اچ‌پی

۱ مقدمه:

ساده‌ترین تعریف از شبکه‌های عصبی، که به صورت کاملتر «شبکه عصبی مصنوعی» نامیده می‌شود، توسط دکتر رابرت هچ-نیلسن ارائه شده است، در این تعریف آمده است:

«... یک سیستم محاسباتی متشکل از تعدادی عنصر پردازشی[4] ساده با قابلیت بالای ارتباط با یکدیگر

که داده‌های ورودی را با توجه با حالت پویای خود پردازش می‌کنند.»[۱]

شبکه‌های عصبی مصنوعی دستگاه‌های پردازشی (سخت‌افزار یا الگوریتم) هستند که با الهام از ساختار قشر مغزی پستانداران در مقیاس بسیار کوچکتر مدل شده‌اند. یک شبکه عصبی مصنوعی با مقیاس بزرگ به طور معمول می‌تواند صدها یا هزاران واحد پردازشی داشته باشد و این در حالی است که مغز پستانداران دارای بیلیون‌ها نورون با قابلیت برقراری ارتباط بیشتر هستند.

شبکه‌های عصبی مصنوعی متشکل از حداقل ۳ لایه و یا بیشتر هستند. لایه‌ها متشکل از چندین گره[5] هستند که با یکدیگر در تماس بوده و هر گره شامل یک تابع فعال‌سازی[6] هست. الگوها توسط لایه ورودی[7] به شبکه ارائه می‌شوند سپس ورودی‌ها به یک یا چندین

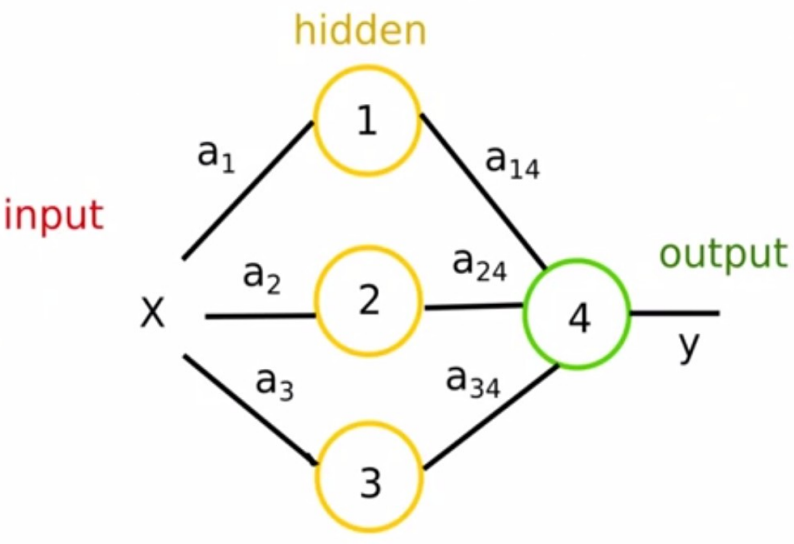
---

۱ پست‌الکترونیکی: ala.falaki@gmail.com

۲ ANN
۳ Rule-Based
۴ Processing Unit
۵ Nodes
۶ Activation Function
۷ Input Layer

لایه پنهان[۸] ارسال خواهند شد و فرآیند پردازش داده‌ها توسط ارتباطات وزن‌دار انجام خواهد شد. در نهایت آخرین لایه پنهان به خروجی متصل می‌شود و خروجی شبکه نمایش داده خواهد شد.

جهت درک بهتر مطالب ارائه شده به شکل شماره ۱ توجه کنید:



شکل شماره ۱

شکل شماره ۱ شمای کلی یک شبکه عصبی مصنوعی ساده می‌باشد که متشکل از ۳ لایه و ۴ عصب[۹] یا همان گره است. رنگ قرمز همان لایه ورودی می‌باشد. (در این مثال تنها یک ورودی وجود دارد که در شکل با اسم x مشخص شده است.) لایه دوم که با رنگ زرد مشخص شده است، لایه پنهان نام دارد. (در این مثال یک لایه پنهان دارای ۳ عصب به تصویر کشیده شده است.) لایه سوم که همان لایه خروجی می‌باشد با رنگ سبز مشخص شده است. (در این مثال یک عصب خروجی مشخص شده است.)

لایه پنهان خود باید حداقل از یک لایه تشکیل شده باشد و می‌تواند با توجه به پیچیدگی مسأله به چندین لایه همتقسیم شود و همچنین هر لایه می‌تواند از چندین عصب تشکیل شده باشد. و همچنین لایه ورودی و خروجی هر کدام یک لایه منحصر به فرد از شبکه هستند که تعداد ورودی و خروجی شبکه را مشخص میکنند.

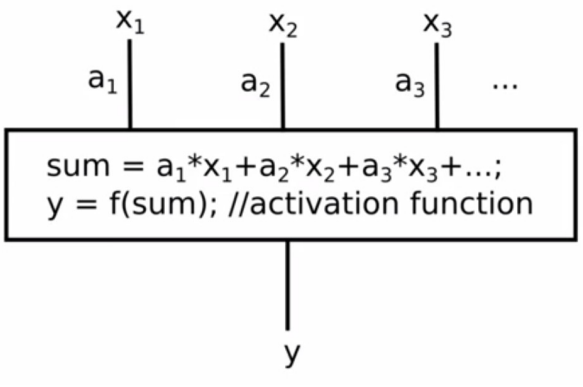
همان‌طور که در شکل شماره ۱ مشاهده می‌کنید هر خط ارتباطی میان نورون‌ها دارای یک وزن می‌باشد. (a1, a2, a3, a14, a24,a34) این وزن‌ها اهمیت هر مسیر را مشخص کرده و همچنین تأثیر مستقیم بر روی خروجی هر نورون دارند به نحوی که با محاسبه وزن درست برای هر مسیر می‌توانیم خروجی‌ای با دقت بالا به دست بیاوریم. در مرحله آموزش[۱۰] (یادگیری[۱۱]) شبکه عصبی با وارد کردن اطلاعات موجود و آزمایش شده که جواب آنها از قبل برای ما مشخص بوده‌اند، اقدام به آموزش شبکه عصبی می‌کنیم. در این مرحله اطلاعات ورودی را به شبکه داده و شبکه پس از محاسبه خروجی، آن‌ها را با خروجی مورد نظر ما مقایسه میکند. در صورت اشتباه بودن خروجی نسبت به خروجی مورد نظر ما، شبکه اقدام به تصحیح وزن‌های هر مسیر می‌کند تا خروجی متناسب با خروجی مورد نظر ما ارائه بدهد.

جهت مشاهده شیوه محاسبه خروجی توسط هر نورون به شکل شماره ۲ توجه کنید.

---

۸ Hidden Layer
۹ Neron
۱۰ Training
۱۱ Learning

در این مثال فرض می‌کنیم یک نورون از ۳ ورودی و یک خروجی تشکیل شده است. همان‌طور که در شکل مشاهده می‌کنید، بعد از ضرب هر ورودی با وزن آن مسیر و جمع اعداد به دست آمده، آنها را وارد تابعی به نام تابع فعال سازی می‌کنیم. این تابع وظیفه پیدا کردن روابط بین نقطه‌های مختلف در هر مجموعه داده‌ای را دارد. جهت مصارف مختلف می‌توان از توابع مختلف، برای مثال توابع خطی[۱۲]، توابع هلالی[۱۳] و ... استفاده کرد. جهت مشاهده انواع توابع می‌توانید به [۲] مراجعه کنید.



شکل شماره ۲

۲ مراحل ساخت یک شبکه عصبی:

با توجه به توضیحات داده شده در رابطه با شبکه‌های عصبی، ساخت این شبکه‌ها و پیدا نمودن یک تابع فعال‌سازی بهینه نیاز به دانش ریاضی بالایی دارد. به همین دلیل جهت آسان نمودن این فرآیند می‌توان از کتابخانه FANN[۱۴] استفاده نمود.

۱.۲ معرفی FANN:

یک کتابخانه رایگان و متن‌باز جهت ساخت شبکه‌های عصبی مصنوعی چند لایه نوشته شده با زبان برنامه‌نویسی C است که توانایی پشتیبانی از شبکه‌های متصل کامل[۱۵] و شبکه‌هایی با اتصال کم[۱۶] را دارد و همچنین بدون وابستگی به پلتفرم خاص[۱۷] عمل می‌کند. این کتابخانه یک چهارچوب جهت آموزش دادن شبکه عصبی با استفاده از داده‌های موجود به شیوه‌ای آسان ارائه داده است و همچنین قابلیت استفاده بر روی ۱۵ زبان برنامه‌نویسی را دارد.[۳]

(در این مقاله کدهای ارائه شده و همچنین شیوه ساخت شبکه بر اساس زبان برنامه‌نویسی PHP ارائه خواهد شد. از قابلیت‌های این چهارچوب توانایی ذخیره سازی مدل ایجاد شده و استفاده از آن مدل در زبان‌های دیگر می‌باشد که در ادامه بیشتر در این باره صحبت خواهد شد.)

۲.۲ نصب FANN بر روی PHP

در این مقاله مراحل نصب بر اساس سیستم عامل Ubuntu عنوان خواهد شد، جهت مشاهده شیوه نصب برای سایر توزیع‌های لینوکس به مستندات این افزونه[۱۸] به آدرس [۴] مراجعه نمایید.

---

۱۲ Linear
۱۳ Sigmoid
۱۴ Fast Artificial Neural Network
۱۵ Fully Connected
۱۶ Sparsely Connected
۱۷ Cross-Platform
۱۸ Extension

الف) ساده‌ترین روش نصب استفاده از PECL است. ابتدا باید اطمینان حاصل نمایید کـه ورژن توسـعه[۱۹] libfann بـر روی سیسـتم عامل شما نصب شده باشد. جهت نصب این بسته دستور زیر را وارد نمایید:

```
$ sudo apt-get install libfann-dev
```

ب) پس از اطمینان از نصب بسته ذکر شده، با استفاده از دستور زیر می‌توانید کتابخانه را به صورت کامل نصب نمایید.

```
$ sudo pecl install fann
```

ج) و در نهایت فایل تنظیمات PHP یا همان php.ini را باز نموده و نام افزونه را مشخص نمایید. جهت باز نمودن فایـل مـورد نظـر در PHP نسخه ۵ می‌توانید دستور زیر را وارد نمایید.

```
$ sudo nano /etc/php5/apache2/php.ini
```

د) و خط زیر را در انتهای فایل درج نمایید.

```
extension=fann.so
```

ه) و در نهایت جهت به روز رسانی PHP دستور زیر را وارد نمایید.

```
$ sudo service apache2 restart
```

در صورتی که مراحل را درست انجام داده باشید، با اجرای تابع phpinfo و جستجوی کلمه FANN اسم این افزونه را مشاهده خواهید کرد.

۳.۲ شیوه کار با کتابخانه

جهت مشاهده یک مثال مقدماتی از شیوه اجرای این کتابخانه و همچنین روش کار آن به آدرس [۵] مراجعه نمایید. در این مثال شیوه پیاده‌سازی یک گیت XOR جهت محاسبه خروجی تک بیتی شرح داده شده است. به طور کل بعد از آنالیز مسـأله و مشـخص نمـودن ورودی و خروجی‌های آن، ساخت چنین شبکه‌هایی در این کتابخانه از ۲ مرحله تشکیل می‌شود. مرحله اول جمع‌آوری اطلاعات مـورد نیاز[۲۰] و مرحله دوم آموزش شبکه با توجه به داده‌های جمع‌آوری شده است. سپس با استفاده از مدل بدست آمـده می‌تـوان خروجـی سوال‌هایی که در مجموعه داده[۲۱] موجود نیست را به دست آورد. جهت مشاهده شیوه ساخت این شبکه‌ها، پروژه نمونه در ادامـه ایـن مقاله ارائه خواهد شد و سعی می‌شود مراحل کار شرح داده شوند.

۳ پروژه نمونه:

در این پروژه نمونه سعی می‌کنیم یک شبکه عصبی مصنوعی جهت تشخیص زبان فارسی و انگلیسی ارائه دهیم. ابتدا سعی می‌کنیم بـا تحلیل مسأله ورودی‌ها و خروجی‌ها را شناسایی نماییم. در این پروژه ۳ متغیر به عنوان ورودی مشخص شده‌اند. وزن حروف فارسـی در رشته، وزن حروف انگلیسی در رشته و وزن حروف تشخیص داده نشده در رشته. (وزن هریک از موارد گفته شده بـا تقسـیم تعـداد آن مورد به کل حروف رشته محاسبه می‌شود.) خروجی‌های این مسأله نیز به ۲ متغیر تقسیم می‌شوند.(متغیر اول احتمـال فارسـی بـودن

---

۱۹ Development
۲۰ Dataset
۲۱ Dataset

رشته، متغیر دوم احتمال انگلیسی بودن رشته.)

در مرحله بعد جهت آموزش شبکه عصبی نیاز به داده‌هایی داریم که خروجی آنها برای ما مشخص باشد. به مثال زیر توجه نمایید:

```
0.4792 0.2708 0.2500
1 0
```

هر مورد از مجموعه داده، از ۲ خط تشکیل می‌شود. خط اول نشان‌دهنده ورودی و خط بعد نشان‌دهنده خروجی شبکه خواهد بود. در مثال ذکر شده خط اول نشان‌دهنده یک رشته است که به ترتیب وزن حروف فارسی آن ۰/۴۷۸۲ ، وزن حروف انگلیسی آن ۰/۲۷۰۷ و وزن حروف تشخیص داده نشده ۰/۲۵۰۰ بوده است. خط دوم این مثال مشخص می‌کند که خروجی مورد انتظار ما از این داده یک رشته فارسی است. (اگر رشته مورد نظر انگلیسی بود، خط دوم به حالت 1 0 نوشته می‌شد.)

شبکه عصبی با خواندن داده‌ها و تولید یک خروجی در هر تکرار، خروجی تولید شده خود را با خروجی مورد انتظار ما مقایسه می‌کند. در صورت وجود تفاوت، شبکه عصبی اقدام به تغییر وزن‌های هر نورون می‌کند، به نحوی که خروجی تولید شده خود را منطبق بر خروجی مورد انتظار ما قرار دهد.

در ساختار کلی یک مجموعه داده، در اولین خط، ۳ عدد مشاهده می‌شود.(این خط جهت راهنمایی کتابخانه FANN و معرفی شیوه خواندن مجموعه داده نوشته می‌شود.) عدد اول تعداد داده‌های موجود (ورودی/خروجی) در مجموعه داده مذکور، عدد دوم نشان‌دهنده تعداد ورودی‌ها و عدد سوم نشان‌دهنده تعداد خروجی‌ها می‌باشد.

مثال: (بخشی از مجموعه داده استفاده شده در این پروژه، ۳ مورد از ۱۷ مورد، برای مشاهده کامل مراجعه شود به [۶])

```
17 3 2
0.4792 0.2708 0.2500
1 0
0.2444 0.4222 0.3333
0 1
0.7826 0.0000 0.2174
1 0
```

مرحله بعد ساخت مدل و آموزش آن براساس مجموعه داده کامل شده است. جهت آموزش شبکه عصبی به کد زیر توجه کنید:

```php
<?php
// Number of inputs
$num_input = 3;
// Number of outputs.
$num_output = 2;
// Number of overall layers. (counting input & output layers)
$num_layers = 3;
// Number of neurons on hidden layer.
$num_neurons_hidden = 3;
// Continue learning process until the error is very small(0.001)
// Or if the iteration passes 500000 times.
$desired_error = 0.001;
$max_epochs = 500000;

$ann = fann_create_standard($num_layers, $num_input, $num_neurons_hidden, $num_output);

if ($ann) {
   fann_set_activation_function_hidden($ann, FANN_SIGMOID_SYMMETRIC);
   fann_set_activation_function_output($ann, FANN_SIGMOID_SYMMETRIC);

   $filename = dirname(__FILE__) . "/input.dataset";
   $train_data = fann_read_train_from_file( $filename );
```

```
    if ( fann_train_on_data($ann, $train_data, $max_epochs, null, $desired_error))
        fann_save($ann, dirname(__FILE__) . "/langDetector.net");

    $mse = fann_get_MSE($ann);
    echo "<p>Your Artificial Neural Network has been created, You can see it in 'langDetector.net' file.</p>";
    echo "<p>The MSE (Mean Squared Error) is: {$mse} (close to zero means better)";

    fann_destroy($ann);
}
```

در این مثال، ابتدا تعداد ورودی، خروجی، تعداد نورون‌های لایه مخفی، خطای مورد انتظار و تعداد تکرار برای رسیدن به جواب بهینه را به ترتیب مقداردهی می‌کنیم. سپس با استفاده از موارد مقداردهی شده و تابع fann_create_standard یک شبکه عصبی ایجاد می‌کنیم. در ادامه نوع تابع فعال‌سازی برای نورون‌های لایه مخفی و نورون‌های خروجی را مشخص می‌نماییم. سپس فایل مجموعه داده را فراخوانی کرده و با استفاده از تابع fann_train_on_data شروع به آموزش شبکه عصبی بر اساس داده‌های مجموعه داده می‌کنیم. در نهایت با استفاده از تابع fann_save مدل بدست آمده را برای استفاده در آینده ذخیره کرده، مقدار خطای مدل را نشان داده و در نهایت مدل بدست‌آمده را از حافظه پاک می‌کنیم.

در مرحله آخر با استفاده از کد زیر می‌توانیم مدل بدست‌آمده را آزمایش نماییم. در این مرحله می‌توانیم با وارد کردن ۳ ورودی که همان وزن هر حرف در رشته می‌باشد، (عدد اول وزن کلمات فارسی، عدد دوم وزن کلمات انگلیسی و عدد سوم وزن کلمات مشخص نشده) مشخص نماییم رشته وارد شده فارسی یا انگلیسی می‌باشد.

در مثال زیر برای آزمایش مدل خود، رشته «شناسایی زبان نوشتاری با استفاده از Artificial Neural Network و کتابخانه FANN» را به عنوان نمونه وارد کرده‌ایم. (رشته مذکور در زمان آموزش مدل وارد نشده بود.) ورودی‌های مدل به ترتیب عدد ۰/۵ برای حروف فارسی، عدد ۰/۳۴۲۱ برای حروف انگلیسی و عدد ۰/۱۵۷۹ برای حروف تشخیص داده نشده است. ورودی‌های بدست‌آمده را به شبکه عصبی می‌دهیم و خروجی مورد نظر که همان زبان نوشتاری رشته است را دریافت خواهیم نمود.

به کدهای زیر دقت نمایید:

```
<?php
$train_file = (dirname(__FILE__) . "/langDetector.net");
if (!is_file($train_file))
    die("The file langDetector.net has not been created! Please run train_model.php to generate it");

$ann = fann_create_from_file($train_file);

if (!$ann)
    die("ANN could not be created");

$input = array( 0.5, 0.3421, 0.1579 );
$calc_out = fann_run($ann, $input);
fann_destroy($ann);
print_r( $calc_out );
```

خروجی کد نوشته شده:

```
Array ( [0] => 0.96449780464172 [1] => -0.035985063761473 )
```

همان‌طور که پیش‌تر توضیح داده شد، اندیس اول آرایه نشان‌دهنده احتمال فارسی بودن زبان رشته و اندیس دوم آرایه نشان‌دهنده احتمال انگلیسی بودن زبان رشته می‌باشد. که این شبکه عصبی به درستی زبان رشته را فارسی پیش‌بینی نمود.

به عنوان نکته آخر توجه داشته باشید، به دلیل استفاده از تابع فعال‌سازی هلالی اعداد انتخابی در مجموعه داده باید در بازه -۱ تا ۱ و

یا در بازه ۰ تا ۱ قرار بگیرند.

تمامی کدهای مورد استفاده جهت اجرای این اسکریپت بر روی گیت‌هاب با مجوز MIT منتشر شده است. برای مشاهده کدها می‌توانید به آدرس [۷] مراجعه نمایید.

توضیح کدهای منتشر شده: در کدهای منتشر شده فایلی تحت عنوان parser.php وجود دارد. وظیفه این فایل تبدیل رشته به وزن‌های هر زبان می‌باشد که هدف از نوشتن این اسکریپت راحتی در محاسبه وزن‌ها جهت ایجاد فایل مجموعه داده بوده است. از همین اسکریپت در فایل test_model.php هم جهت وارد کردن رشته آزمایشی و مشاهده خروجی استفاده شده است.

۴ نتیجه گیری:

در این مقاله ابتدا مقدمه‌ای از شبکه‌های عصبی مصنوعی و شیوه کار آنها ذکر شد، در ادامه کتابخانه‌ای جهت ساخت اینگونه شبکه‌های عصبی معرفی شد و در نهایت سعی شد با معرفی پروژه‌ای کمی پیچیده‌تر از پروژه موجود بر روی مستندات سایت PHP اقدام به توضیح شیوه کار این کتابخانه شود. بدون تردید پروژه مذکور از دید برنامه‌نویسان هر زبان با استفاده از چند شرط و حلقه به سادگی قابل پیاده‌سازی خواهد بود و شاید نیازی به استفاده شبکه‌های عصبی مصنوعی برای چنین مسئله‌ای حس نشود. اما فراموش نکنید که ابتدایی‌ترین هدف از این مقاله آموزش مقدماتی این کتابخانه بوده، همچنین درک شیوه ساخت مجموعه‌داده‌ها، طریقه آموزش مدل و استفاده از مدل در پروژه بوده است. به طور قطع اگر صورت مسأله از شناسایی بین دو زبان نوشتاری، به چهار زبان نوشتاری تغییر کند، در زمان شناسایی این زبان‌ها با استفاده از روش‌های سنتی، تعداد شرط‌ها و حلقه‌ها چند برابر خواهند شد. پس توجه نمایید هدف از این مقاله توانایی اضافه کردن قابلیت تصمیم‌گیری (بدون استفاده از برنامه‌نویسی قانون محور) به نرم‌افزارها بوده است.

طراحی اسکریپت شناسایی اسپم، پیاده‌سازی بازی، پردازش زبان طبیعی و به طور کلی توانایی مدل‌سازی داده‌ها براساس الگو قابل استخراج مجموعه داده‌ها از قابلیت‌های استفاده از هوش مصنوعی هست.

از مزایای استفاده از این روش قابلیت انتقال ساده نرم‌افزارهای نوشته شده بین زبان‌های متفاوت می‌باشد. این کتابخانه قابلیت استفاده بر روی زبان‌هایی مانند JS، NodeJS، Python، Java و بسیاری زبان دیگر را دارد. پس تنها با دانستن منطق کلی استفاده از این کتابخانه به راحتی می‌توانید در زبان‌های مختلف از آن استفاده نمایید.

۵ منابع:

[۱] "Neural Network Primer: Part I" by Maureen Caudill, AI Expert, Feb. 1989

[۲] https://en.wikipedia.org/wiki/Activation_function

[۳] https://sourceforge.net/projects/fann/

[۴] http://php.net/manual/en/fann.installation.php

[۵] http://fann.sourceforge.net/fann_en.pdf

[۶] https://github.com/AlaFalaki/ANN-languageDetecor/blob/master/input.dataset

[۷] https://github.com/AlaFalaki/ANN-languageDetecor