



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ی مهندسی کامپیوتر و فناوری اطلاعات

ماشین-فاکتورگیری مبنی بر شبکه عصبی برای پیش بینی CTR (DeepFM)

نگارش:

زهره دهقانیان

ملیکا عبداللهی

یاسمن میرمحمد

استاد راهنما:

دکتر مریم امیرمزلقانی

بهمن ۱۳۹۷

فراگیری ویژگی های پیچیده فعل و انفعال و تعامل در پشت رفتارهای کاربر برای به حداکثر رساندن CTR برای سیستم پیشنهاد کننده بسیار مهم است. با وجود پیشرفت های قابل توجه، به نظر میرسد روش های موجود تعصب قوی نسبت به کنش های کم مرتبه و یا مرتبه بالا، یا نیاز به تخصص مهندسی ویژگی دارند. در این مقاله، ما نشان می دهیم که ممکن است یک مدل یادگیری انتها به انتها که بر روی تعامل های ویژگی مرتبه پایین و مرتبه بالا تاکید دارد، نتیجه گیری شود. مدل پیشنهادی DeepFM، توان ماشین های فاکتور سازی را برای پیشنهاد و یادگیری deep برای یادگیری ویژگی در یک معماری شبکه عصبی جدید ترکیب می کند. در مقایسه با آخرین مدل گسترده و deep از Google، DeepFM دارای ورودی مشترک به قسمت های "گسترده" و "deep"، بدون نیاز به مهندسی ویژگی در کنار ویژگی های خام است. آزمایش های ادراکی برای نشان دادن اثربخشی انجام می شود و بهره وری DeepFM بر روی مدل های موجود برای پیش بینی CTR، در هر دو معیار داده ها و داده های تجاری است.

1.مقدمه.....	5
2.شیوه و رویکرد.....	7
Deep FM2_1.....	7
۲_۲_۱ محتویات FM.....	8
۲_۲_۲ جز Deep.....	9
۲_۲ ارتباط با دیگر شبکه های عصبی	11
۳.آزمایشات	12
۳_۱ تنظیمات آزمایشی	۱۳
۳_۱_۱ مجموعه داده ها.....	13
۳_۱_۲ معیار های ارزیابی.....	13
۳_۱_۳ مقایسه مدل.....	13
۳_۱_۴ تنظیمات پارامتری.....	13
۳_۲ ارزیابی عملکرد.....	14
۳_۲_۱ مقایسه کارایی.....	14
۳_۲_۲ مقایسه اثربخشی.....	14
۳_۳ مطالعه با حداکثر پارامتر.....	14
۳_۳_۱ مقایسه کارایی.....	16
۳_۳_2 خروجی.....	16
۳_۳_3 تعداد نورون ها در لایه.....	17
۳_۳_۴ تعداد لایه های مخفی.....	17
۳_۳_۵ شکل شبکه.....	18

۴. کار مرتبط 18

۵. نتیجه گیری 19

۶. فهرست مراجع 19

فهرست اشکال

شکل ۱ ۵

شکل ۲ ۸

شکل ۳ ۹

شکل ۴ ۱۰

شکل ۵ ۱۱

جدول ۱ ۱۲

شکل ۶ ۱۴

جدول ۲ ۱۵

شکل ۷ ۱۵

شکل ۸ ۱۷

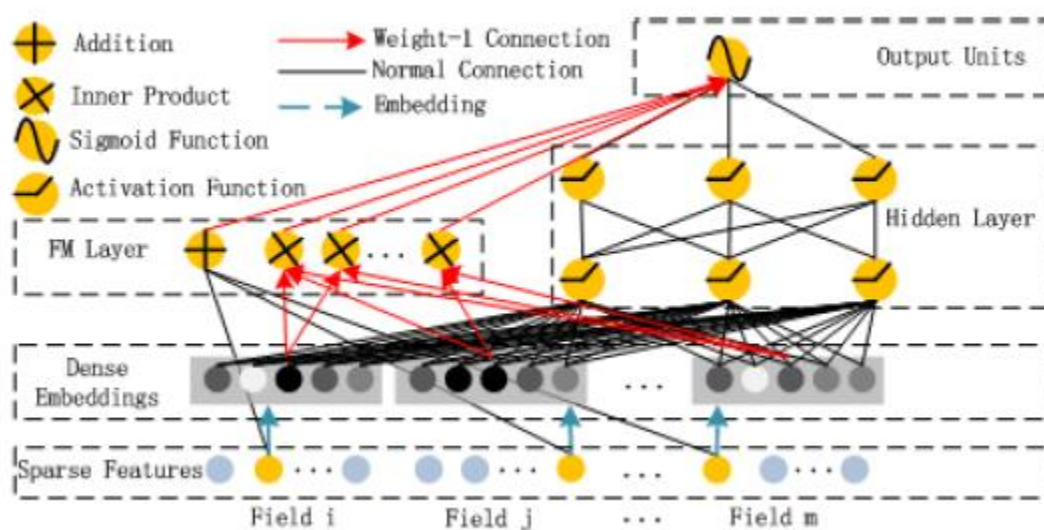
شکل ۹ ۱۷

شکل ۱۰ ۱۸

شکل ۱۱ ۱۸

۱. مقدمه

پیش بینی نرخ کلیک نسبت به رتبه (CTR) در سیستم پیشنهاد دهنده بسیار مهم است. جایی که وظیفه برآورد احتمال یک کاربر روی یک مورد توصیه شده کلیک میکند، است. در بسیاری از سیستم های پیشنهاد کننده هدف به حداکثر رساندن تعداد کلیک است، و به همین ترتیب آیتم بازگشتی به یک کاربر می تواند بر اساس CTR برآورد شده رتبه بندی شود. در حالی که در سایر سناریوهای کاربردی مانند تبلیغات آنلاین نیز افزایش درآمد و بهبود سود مهم است و بنابراین استراتژی رتبه بندی می تواند به عنوان پیشنهاد $CTR \times bid$ در میان همه نامزدها تنظیم شود، که در آن "bid" سودی است که سیستم اگر توسط یک کاربر کلیک شده باشد، دریافت می کند. در هر صورت، روشن است که کلید در برآورد CTR به درستی است.



شکل ۱: معماری گسترده و DeepFM deep. گسترده و deep جزء همان ورودی ورودی خام را به اشتراک می گذارد که امکان پذیر است DeepFM برای یادگیری تداخلات ویژگی پایین و مرتبه بالا به طور همزمان از ویژگی های خام ورودی.

برای پیش بینی CTR مهم است که ویژگی مجازی فعل و انفعالات در پشت پرده رفتارهای کلیک کاربر یاد بگیریم. با مطالعات ما در بازار داغ اپلیکیشن ها، متوجه شدیم که مردم اغلب برنامه های تحویل غذا را در زمان وعده های غذایی ناهار و شام دانلود می کنند، این نتیجه نشان می دهد که (سفارش -۲) کنش های بین دسته بندی و گروه برنامه و زمان می تواند به عنوان یک سیگنال برای CTR استفاده شود. به عنوان مشاهده دوم، نوجوانان پسر بازی های تیراندازی و بازی های RPG را بیشتر می پسندند، به عنوان فعل و

انفعال دسته بندی اپلیکیشن ها (سفارش - ۳) جنسیت کاربر و سن می تواند سیگنال دیگری برای CTR باشد. به طور کلی، چنین کنش های از ویژگی ها پشت رفتار کلیک کردن کاربران می تواند بسیار پیچیده باشد، که در آن هر دو تعامل ویژگی های پایین و بالا مرتبه باید نقش مهمی بازی کند. با توجه به بینش از مدل Wide & Deep [چنگ و همکاران، ۲۰۱۶] از گوگل، که به طور همزمان کنش های با ویژگی های پایین و بالا مرتبه را مورد بررسی قرار می دهد بهبود بیشتری را در مورد موارد ذکر شده و یا به تنهایی به ارمغان می آورد.

با وجود سادگی آن ها، مدل های خطی تعمیم یافته مانند FTRL [McMahan و همکاران، ۲۰۱۳]، عملکرد مناسب را در عمل نشان داده اند. با این وجود، یک مدل خطی، توانایی یادگیری از کنش های ویژگی را ندارد و یک تمرین معمول، شامل شدن کنش های ویژگی متقارن در بردار ویژگی آن است. چنین روشی برای تعمیم دادن به مدل کنش های ویژگی high-order یا مواردی که هرگز یا به ندرت در داده های آموزشی ظاهر نمی شود، است [2010, Rendle]. ماشین های فاکتورسازی [Rendle (FM)]، [2010] کنش های ویژگی متقارن را به عنوان محصل درونی از بردارهای پنهان بین ویژگی ها مدل سازی می کند و همچنین نتایج بسیار مثبتی را نشان می دهد. در حالی که در اصل FM می تواند کنش های ویژگی بالا را مدل سازی کند، در عمل معمولاً فقط کنش های ویژگی (2 - order) به دلیل پیچیدگی های بالا در نظر گرفته می شوند.

به عنوان یک رویکرد قدرتمند برای نشان دادن ویژگی های یادگیری، شبکه های عصبی deep پتانسیل و توانایی یادگیری کنش های پیچیده را دارند. بعضی از ایده ها CNN و RNN را برای CTR توسعه می دهند [Liu et al., 2015، ژانگ و همکاران، ۲۰۱۴]، اما مدل های مبتنی بر CNN به کنش های بین ویژگی های همسایه تعصب دارند، در حالی که مدل های مبتنی بر RNN برای داده های کلیک با وابستگی پیوسته مناسب تر هستند. [Zhang و همکاران، ۲۰۱۶] نمایش ویژگی را مطالعه می کنند و پیشنهاد می کند که ماشین فاکتورسازی شبکه عصبی (FNN) را پشتیبانی می کند. این مدل قبل از اعمال DNN و FM را آموزش می دهد، بنابراین توسط قابلیت های FM محدود می شود. تعامل ویژگی در [2016, Qu et al.] مورد بررسی قرار گرفته است، در این بررسی معرفی یک لایه محصل بین لایه جاسازی شده و لایه کاملاً متصل و ارائه یک شبکه عصبی مبتنی بر محصل (PNN) صورت پذیرفته است. همانطور که در [چنگ و همکاران، ۲۰۱۶] اشاره شد، PNN و FNN، مانند سایر مدل های deep، چندین تعامل ویژگی پایین را دریافت می کنند که برای پیش بینی CTR ضروری است. [2016, Cheng et al.] برای مدل سازی هر دو تعامل ویژگی های پایین و بالا مرتبه، یک ساختار شبکه جالب ترکیبی (Wide & Deep) را پیشنهاد می کند که ترکیبی از یک مدل خطی (گسترده) و یک مدل deep است. در این مدل، دو ورودی متفاوت برای "بخش وسیع" و "بخش deep" مورد نیاز است، و ورودی "بخش وسیع" هنوز هم بر روی مهندسی ویژگی های تخصص متکی است.

چیزی که می توان مشاهده نمود این است که مدل های موجود، نسبت به تعامل ویژگی low یا high-order تعصب دارد یا به مهندسی ویژگی متکی هستند. در این مقاله نشان می دهیم که امکان نتیجه گرفتن یک مدل یادگیری که قادر است کنش های ویژگی از تمام order ها را در رفتار انتها-به-انتها، بدون هیچ گونه مهندسی ویژگی علاوه بر ویژگی های خام یاد بگیرد. مشارکت اصلی ما به شرح زیر است:

- ما یک مدل شبکه عصبی جدید DeepFM (شکل ۱) ارائه می دهیم که معماری های FM و شبکه های عصبی deep (DNN) را ادغام می کند. این مدل کنش های low-order نظیر FM و مدل های کنش های ویژگی high-order مانند DNN را مدل سازی می کند. بر خلاف مدل گسترده و deep [چنگ و همکاران، ۲۰۱۶]، DeepFM می تواند بدون هیچ گونه مهندسی ویژگی آموزش انتها-به-انتها داده شود.

- DeepFM را می توان به طور موثر آموزش داده شود، زیرا قسمت وسیع و بخش deep آن، بر خلاف Cheng و همکاران، ۲۰۱۶]، یک ورودی مشابه و همچنین بردار تعبیه را به اشتراک می گذارند. در [چنگ و همکاران، ۲۰۱۶]، بردار ورودی می تواند اندازه های بزرگ باشد زیرا به صورت دستی شامل کنش های ویژگی متقارن طراحی شده درون بردار ورودی از بخش وسیع است که همچنین پیچیدگی آن را بسیار افزایش می دهد.

- ما DeepFM را بر روی داده های معیار و داده های تجاری ارزیابی می کنیم، که نشان دهنده بهبود سازگاران نسبت به مدل های موجود برای پیش بینی CTR است.

۲. شیوه و رویکرد

فرض کنید مجموعه داده برای آموزش شامل n نمونه (x, y) است، که در x یک داده ثبت شده m -fields است که معمولاً شامل یک جفت کاربر و آیتم و $y = \{0, 1\}$ مرتبط با برچسب است که رفتار کلیک کردن کاربر را نشان می دهد ($y = 1$ به این معنی است که کاربر بر روی آیتم کلیک کرده و $y = 0$ یعنی کلیک نکرده است). x ممکن است شامل زمینه های دسته بندی (مثلاً جنسیت، موقعیت مکانی) و زمینه های پیوسته (مثلاً سن) باشد. هر فیلد قطعی به عنوان یک بردار از رمزگذاری یکپارچه یا یک از رمزگذاری پیوسته پس از تقسیم بندی نمایش داده می شود. سپس هر نمونه به $(y; x)$ تبدیل می شود که $x = [xfield1, xfield2, \dots, xfieldj, \dots, xfieldm]$ یک بردار d -بعدی با وجود $xfieldj$ است که نشان دهنده بردار میدان j ام است. به طور معمول، x بعد بالاست زیاد و بسیار ضعیف است. وظیفه پیش بینی

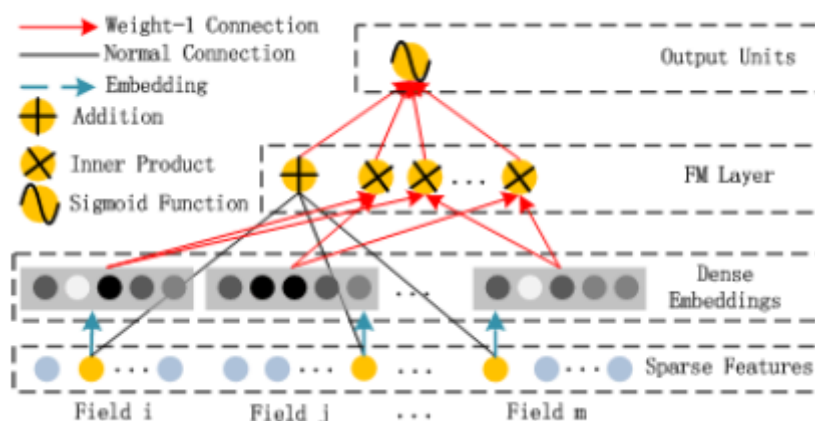
CTR ساخت یک مدل پیش بینی $\hat{y} = \text{CTR model}(x)$ برای برآورد احتمال یک کاربر برای کلیک بر روی یک برنامه خاص در یک زمینه خاص است.

۲,۱ DeepFM

هدف ما یادگیری هر دو تعامل ویژگی های order پایین و بالا است. برای این منظور، ما یک شبکه عصبی مبتنی بر ماشین فاکتورسازی را (DeepFM) پیشنهاد می کنیم. همانطور که در شکل ۱۱ نشان داده شده است، DeepFM شامل دو جزء، مولفه FM و جزء deep است که یک ورودی را به اشتراک می گذارند. برای ویژگی i ، از اسکالر w_i استفاده می شود تا اهمیت $\text{order } 1$ را نشان دهد، یک بردار V_i پنهان برای اندازه گیری تأثیر کنش های آن با سایر ویژگی ها استفاده می شود. V_i در مولفه FM به منظور مدل سازی نظم $\text{order}-2$ و در مولفه های deep برای مدل سازی کنش های ویژگی high-order تغذیه می شود. تمام پارامترها، از جمله w_i, V_i و پارامترهای شبکه $(W(l), b(l))$ در زیر) برای مدل پیش بینی ترکیبی مشترک آموزش داده می شوند:

$$\hat{y} = \text{sigmoid}(y_{FM} + y_{DNN}),$$

که $y = \{0, 1\}$ م CTR پیش بینی شده است، y_{FM} خروجی مولفه FM و y_{DNN} خروجی مولفه deep است.



شکل ۲: معماری FM.

۱-۲-۱ محتویات FM

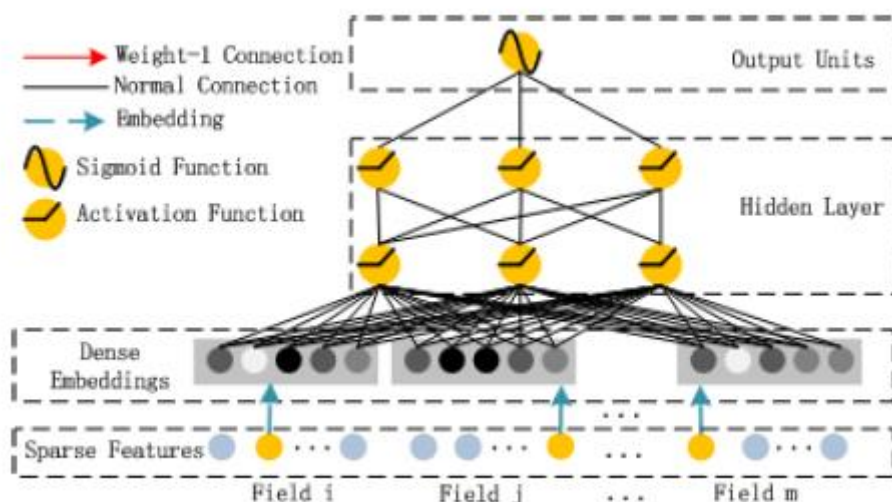
مولفه FM یک ماشین فاکتور سازی است که در [2010, Rendle] پیشنهاد شده است تا کنش های ویژگی را برای توصیه یاد بگیرند. علاوه بر یک تعامل خطی (order-1) بین ویژگیها، مدل های FM به صورت دوبعدی (order-2) کنش های را به عنوان محصول درونی از بردارهای پنهان مربوطه متمایز می کنند.

FM می تواند کنش های order-2 را بسیار موثر تر از رویکردهای قبلی، به ویژه هنگامی که مجموعه داده ها کمیاب است، جذب کند. در رویکردهای قبلی، پارامتر تعامل ویژگی های i و j را می توان تنها زمانی که ویژگی i و ویژگی j هر دو در همان رکورد داده ظاهر می شود. در حالی که در FM، از طریق محصول درونی بردارهای پنهان V_i و V_j اندازه گیری می شود. با تشکر از این طراحی انعطاف پذیر، FM می تواند هر بار که i (یا j) در یک رکورد داده نمایش داده می شود، بردار پنهان V_i (V_j) را آموزش دهد. بنابراین، کنش های ویژگی، که هرگز یا به ندرت در داده های آموزشی ظاهر می شود، توسط FM بهتر یاد گرفته می شود.

همانطور که در شکل ۲ نشان داده شده است، خروجی FM، جمع یک واحد افزوده و تعدادی از واحد های داخلی درونی است:

$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^d \sum_{j_2=j_1+1}^d \langle V_{i_1}, V_{j_2} \rangle x_{j_1} \cdot x_{j_2}, \quad (2)$$

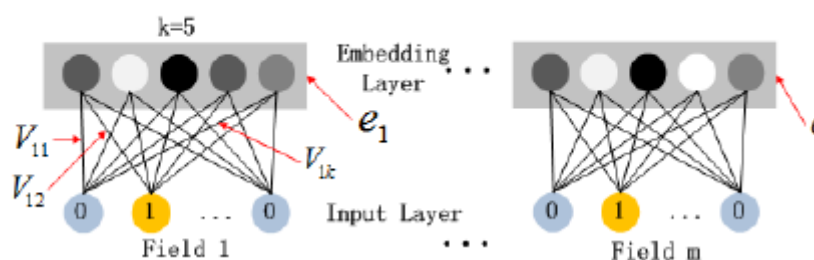
جایی که $w \in \mathbb{R}^d$ و $V_i \in \mathbb{R}^k$ (k داده شده است). واحد افزودنی $(x_i; hw)$ نشان دهنده اهمیت ویژگی‌های order-1 است و واحدهای درونی محصول نشان دهنده تاثیر کنش‌های order-2 می‌باشد.



شکل ۳: معماری DNN.

۲-۱-۲ جزء deep

جزء deep یک شبکه عصبی feed-forward (کنترل پیشخور) است که برای یادگیری کنش‌های high-order استفاده می‌شود. همانطور که در شکل ۳ نشان داده شده است، یک داده ثبت شده (یک بردار) به شبکه عصبی تغذیه می‌شود. در مقایسه با شبکه‌های عصبی با تصویر [He et al., 2016] یا صدا [Boulanger-Lewandowski et al., 2013] به عنوان ورودی که صرفاً مداوم و متراکم است، ورودی پیش بینی CTR کاملاً متفاوت است، که نیازمند یک طراحی معماری شبکه جدید است. به طور خاص، بردار ورودی خام ویژگی برای پیش بینی CTR معمولاً بسیار پراکنده، دارای ابعاد فوق العاده بالا، ترکیبی قطعی - ادامه دار، و در زمینه‌های (به عنوان مثال، جنسیت، محل، سن) دسته بندی شده است. این یک لایه جاسازی کننده برای فشرده سازی بردار ورودی به یک بردار ارزش - واقعی با ابعاد کم متراکم قبل از اضافه شدن به اولین لایه مخفی را نشان می‌دهد، در غیر این صورت شبکه می‌تواند برای آموزش سخت باشد.



شکل ۴: ساختار لایه جاسازی

شکل ۴، ساختار زیر شبکه از لایه ورودی به لایه جاسازی را برجسته می کند. ما می خواهیم دو ویژگی جالب این ساختار شبکه را ذکر کنیم: (۱) در حالی که طول بردارهای میدان ورودی مختلف می تواند متفاوت باشد، جفت های آنها از یک اندازه (k) هستند. (۲) بردارهای ویژگی پنهان (V) در FM در حال حاضر به عنوان وزن شبکه کمک کند که یاد گرفته شده و برای فشرده سازی بردارهای زمینه ورودی به بردارهای تعبیه مورد استفاده قرار گرفته شده است. در [ژانگ و همکاران، ۲۰۱۶]، V توسط FM پیش آموزش دیده است و به عنوان نمونه اولیه استفاده شده است. در این کار، به جای استفاده از بردارهای ویژگی پنهان از FM برای راه اندازی شبکه همانگونه که در [Zhang و همکاران، ۲۰۱۶] انجام شده است، ما مدل FM را به عنوان بخشی از معماری یادگیری کلی مان، علاوه بر مدل DNN دیگر شامل می کنیم. به همین ترتیب، ما نیاز به پیش آموزش توسط FM را از بین می بریم و به جای آن، به طور مشترک به طور کلی شبکه را به صورت تمام وقت آموزش می دهیم.

خروجی لایه جاسازی را به عنوان:

$$a^{(0)} = [e_1, e_2, \dots, e_m], \quad (3)$$

جایی که e_i فیلد i ام است و m تعداد رشته ها است. سپس، a به توان (\cdot) به شبکه عصبی deep تغذیه می شود و فرایند رو به جلو است:

$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)}), \quad (4)$$

جایی که L عمق لایه است و یک تابع فعال است. $a^{(l)}$ ، $w^{(l)}$ ، $b^{(l)}$ خروجی، وزن مدل و انحراف لایه l است. پس از آن، یک بردار ویژگی حقیقی متراکم تولید می شود که در نهایت به تابع sigmoid برای پیش بینی CTR وارد می شود: $y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1})$ که $|H|$ تعداد لایه های پنهان است.

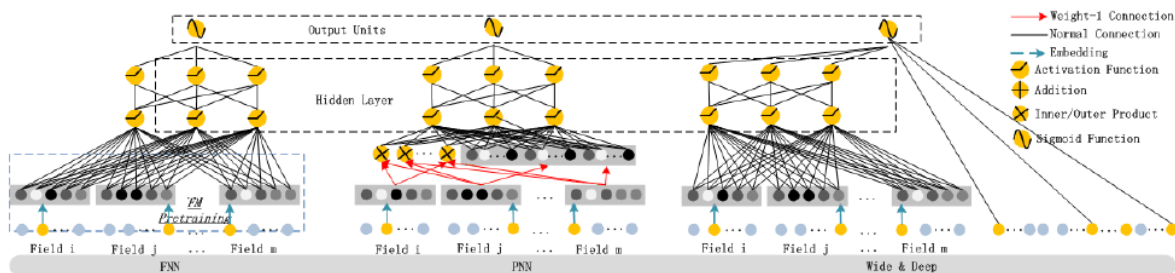
لازم به ذکر است که مولفه FM و جزء deep یک ویژگی جادویی مشابه را به اشتراک می گذارند که دو مزیت مهم را به ارمغان می آورد: (۱) هر دو تعامل ویژگی های پایین و بالا مرتبه را از ویژگی های خام یاد می گیرد؛ (۲) نیازی به مهندسی ویژگی ورودی، به عنوان یک موضوع مورد نیاز در Wide & Deep [چنگ و همکاران، ۲۰۱۶] نیست.

۲_۲ ارتباط با دیگر شبکه های عصبی

با الهام از موفقیت فراوان یادگیری deep در برنامه های مختلف، اخیراً چندین مدل deep برای پیش بینی CTR ایجاد شده است. این بخش DeepFM پیشنهاد شده را با مدل های deep موجود برای پیش بینی CTR را مقایسه می کند.

FNN: همانطور که شکل ۵ (سمت چپ) نشان می دهد، FNN یک شبکه عصبی پیشفرض FM است. [2016, Zhang et al.]. استراتژی پیش آموزش FM منجر به دو محدودیت می شود: (۱) پارامترهای تعبیه شده ممکن است توسط FM تحت تاثیر قرار گیرد؛ (۲) بهره وری توسط سربار معرفی شده توسط مرحله قبل از آموزش کاهش می یابد. علاوه بر این، FNN فقط کنش های با درجه بالا را جذب می کند. در مقابل، در DeepFM نیاز به پیش آموزش نیست و هر دو کنش های high- و low-order را فرا می گیرد/

PNN: برای رسیدن به کنش های با ویژگی های بالا، PNN یک لایه محصول را بین لایه جاسازی و اولین لایه مخفی ایجاد می کند [2016, Qu et al.]. با توجه به انواع مختلف عملیات محصول، سه نوع وجود دارد: IPNN، OPNN، و PNN، که IPNN بر اساس محصول داخلی بردارها است، OPNN بر اساس محصول بیرونی است و PNN بر اساس محصولات داخلی و خارجی است.



شکل ۵: معماری مدل های deep موجود برای پیش بینی FNN، CTR، PNN، مدل گسترده و deep

جدول ۱: مقایسه مدل های deep برای پیش بینی CTR

بدون پیش آموزش	ویژگی High-order	ویژگی Low-order	بدون مهندسی ویژگی	
X	O	X	O	FNN
O	O	X	O	PNN
O	O	O	X	Wide & Deep
O	O	O	O	DeepFM

جدول ۱

برای ارتقا هر چه بیشتر محاسبات، نویسندگان محاسبات تقریبی درونی و بیرونی محصولات را ارائه دادند: (۱) محصول درونی با حذف برخی نورون ها به صورت تقریبی محاسبه می شود؛ (۲) محصول بیرونی توسط فشرده سازی بردارهای ویژگی m ، کی k بعدی به یک بردار k بعدی به صورت تقریبی محاسبه شده است. با این حال، ما متوجه می شویم که محصول بیرونی نسبت به محصول درونی کمتر قابل اعتماد است، زیرا محاسبات تقریبی محصول بیرونی اطلاعات زیادی را از دست می دهد که موجب ناپایداری نتایج می شود. اگرچه محصول درونی قابل اعتمادتر است، اما همچنان از پیچیدگی محاسباتی بالا رنج می برد، زیرا خروجی لایه محصول به تمام نورونهای اولین لایه مخفی متصل است. برخلاف PNN، خروجی از لایه محصول در DeepFM فقط به لایه نهایی خروجی (یک نورون) متصل می شود. مانند FNN، تمام PNN ها کنش های با درجه پایین را نادیده می گیرند.

Wide & Deep: گسترده و deep: Wide & Deep: گسترده و deep (شکل ۵ (سمت راست)) توسط گوگل برای مدل تعامل ویژگی های پایین و بالا مرتبه به صورت همزمان پیشنهاد می شود. همانطور که در [چنگ و همکاران، ۲۰۱۶] نشان داده شده است، نیاز به تخصص مهندسی ویژگی در ورودی به قسمت "گسترده" (به عنوان مثال، محصول برنامه های نصب کاربران و برنامه های تصحیح در توصیه برنامه) وجود دارد. در مقابل، DeepFM نیازی به دانش تخصص خاصی برای اداره ورودی با یادگیری مستقیم از ویژگی های خام ورودی ندارد.

یک فرمت مستقیم برای این مدل LR را توسط FM جایگزین می کند (ما همچنین این توسعه را در بخش ۳ ارزیابی می کنیم). این تعمیم مشابه DeepFM است، اما DeepFM، ویژگی تعبیه بین FM و جزء deep را به اشتراک میگذارد. این استراتژی اشتراک گذاری ویژگی های تعبیه کردن تأثیرات نمایندگی ویژگی های هر دو پایین و تداخل ویژگی های بالا مرتبه، که دقیق تر نمایانگر را نشان می دهد.

روش خلاصه ها: خلاصه، رابطه بین DeepFM و سایر مدل های deep در چهار جنبه در جدول ۱ ارائه شده است. همانطور که مشاهده می شود، DeepFM تنها مدلی است که نیاز به هیچ پیش آموزش و مهندسی ویژگی ندارد، و هر دو تعامل ویژگی های low- و high-order را ضبط می کند.

۳. آزمایشات

در این بخش، ما به مقایسه DeepFM پیشنهادی خودمان و مدل های دیگر مدل های جدید تجربی می پردازیم. نتیجه ارزیابی نشان می دهد که DeepFM پیشنهادی ما موثرتر از هر مدل دیگر جدید و آپدیت شده است و بهره وری DeepFM می تواند با بهترین انواع موجود قابل مقایسه است.

۳-۱ تنظیمات آزمایشی

۳-۱-۱ مجموعه داده ها

ما اثربخشی و کارایی DeepFM پیشنهادی خودمان را در دو مجموعه داده های زیر ارزیابی می کنیم.

(۱) Criteo Dataset: بخش Criteo Dataset شامل داده های ثبت شده از کلیک ۴۵ میلیون کاربر است. ۱۳ ویژگی پیوسته و ۲۶ طبقه بندی وجود دارد. ما مجموعه داده ها را به طور تصادفی به دو قسمت تقسیم می کنیم: ۹۰٪ برای آموزش و تنها ۱۰٪ برای آزمایش است.

(۲) Dataset Company: برای بررسی عملکرد DeepFM در پیش بینی واقعی CTR صنعتی، ما بر روی مجموعه داده های شرکت آزمایشاتی را انجام می دهیم. ۷ روز متوالی رکوردهای کلیک کاربران را از مرکز بازی اپ استور برای آموزش را جمع آوری می کنیم و ۱ روز بعد را نیز برای آزمایش بر روی آن ها قرار داده ایم. حدود ۱ میلیارد پرونده در کل مجموعه داده جمع آوری شده وجود دارد. در این مجموعه داده ها، ویژگی های برنامه (مانند شناسایی، دسته بندی، و غیره)، ویژگی های کاربر (به عنوان مثال، برنامه های دانلود شده کاربر و غیره) و ویژگی های زمینه (مثلا زمان عملیات و غیره) وجود دارد.

۳-۱-۲ معیارهای ارزیابی

ما از دو معیار ارزیابی در آزمایشات استفاده می کنیم: AUC (Area Under ROC) و Logloss (آنترپی متقاطع).

۳_۱_۳ مقایسه مدل

ما ۹ مدل در آزمایش هایمان را مقایسه می کنیم: LR، FM، FNN، PNN (سه نوع)، Wide & Deep، و DeepFM. در مدل Wide & Deep، به منظور از بین بردن تلاش های مهندسی ویژگی، ما از مدل اصلی Wide & Deep با جایگزینی LR توسط FM به عنوان بخش گسترده استفاده می کنیم. به منظور تشخیص این دو نوع از Wide & Deep، ما آنها LR & DNN و FM و DNN، به ترتیب می نامیم.

۳_۱_۴ تنظیمات پارامتری

برای ارزیابی مدل ها در مجموعه داده Criteo ما از تنظیمات پارامتر در [Qu و همکاران، ۲۰۱۶] برای FNN و PNN پیروی می کنیم:

(۱) خروج: ۵، ۰؛ (۲) ساختار شبکه: ۴۰۰-۴۰۰-۴۰۰؛ (۳) بهینه ساز: adam؛ (۴) تابع فعال سازی: tanh برای IPNN و relu برای دیگر مدل های deep.

برای رعایت عدالت، DeepFM پیشنهادی ما از همان تنظیمات استفاده می کند. بهینه سازان LR و FM به ترتیب FTRL و Adam هستند و بعد پنهان، FM 10 است. برای دستیابی به بهترین عملکرد برای هر مدل فردی در مجموعه داده های شرکت، ما یک مطالعه دقیق پارامتری انجام دادیم که در بخش ۳،۳ بحث شده است.

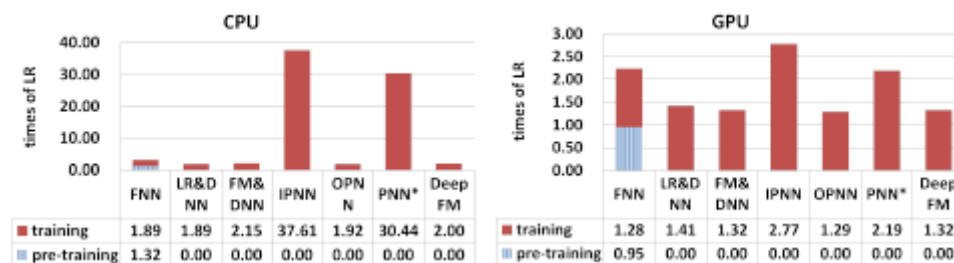
۳_۲ ارزیابی عملکرد

در این بخش، مدل های ذکر شده در بخش ۳،۱ در دو مجموعه داده ها را برای مقایسه کارایی و بهره وری آنها ارزیابی می کنیم.

۳_۲_۱ مقایسه کارایی

کارایی مدل های یادگیری deep برای اپلیکیشن های واقعی جهان بسیار مهم است. نتایج نشان داده شده در شکل ۶ شامل تست های CPU (سمت چپ) و GPU (سمت راست) است، جایی که مشاهدات زیر را

داریم: (۱) پیش - آموزش از FNN کارآمدی آن را کمتر می کند؛ (۲) هرچند سرعت پردازش IPNN و PNN در پردازنده گرافیکی بالاتر از سایر مدل ها است، اما به دلیل عملیات درونی محصول ناکارآمد، هنوز از نظر هزینه محاسباتی گرانقیمت است. (۳) DeepFM تقریباً در هر دو آزمایش، کارآمدتر است.



شکل ۶) مقایسه زمان

۲-۲-۳ مقایسه اثر بخشی

عملکرد برای پیش بینی CTR مدل های مختلف در مجموعه داده Criteo و مجموعه داده های شرکت در جدول ۲ نشان داده شده است.

جایی که مشاهدات زیر را داریم:

- کنش های ویژگی یادگیری عملکرد مدل پیش بینی CTR را بهبود می بخشد. این مشاهدات از این واقعیت است که LR (که تنها مدل است که کنش های را در نظر نمیگیرد) بدتر از سایر مدلها عمل میکند. به عنوان بهترین مدل، عملکرد خارجی DeepFM با ۰٫۸۶٪ و ۴٫۱۸٪ از AUC نسبت به LR (۱٫۱۵٪ و ۵٫۶۰٪ در شرایط Logloss) در مجموعه داده های شرکت و Criteo است.

- یادگیری کنش های ویژگی high- و low-order به طور همزمان و به طور صحیح عملکرد مدل پیش بینی CTR را بهبود می بخشد. عملکرد خارجی DeepFM از مدلهایی که تنها کنش های با درجه پایین نظیر (FM) یا کنش های با ویژگی بالا (یعنی FNN، IPNN، OPNN، PNN) را یاد میگیرد بهتر عمل میکند. در مقایسه با بهترین مدل دوم، DeepFM بیش از ۰٫۳۷٪ و ۰٫۲۵٪ از نظر AUC (۰٫۴۲٪ و ۰٫۲۹٪ از نظر Logloss) در مجموعه داده های شرکت و Criteo بدست می آورد.

- یادگیری کنش های ویژگی high- و low-order همزمان در حالی که ویژگی های مشابه همانند سازی برای یادگیری کنش های ویژگی اشتراک گذاری high- و low-order عملکرد مدل پیش بینی CTR را بهبود می بخشد. DeepFM بهتر از مدل هایی که کنش های ویژگی high- و low-order را یاد می گیرند عملکرد بهتری دارد. این مدل ها از ویژگی جداگانه (یعنی LR & DNN و FM و DNN) استفاده می کنند. در مقایسه با این دو مدل DeepFM بیش از ۰.۴۸٪ و ۰.۳۳٪ از نظر AUC (۰.۶۱٪ و ۰.۶۶٪ در شرایط Logloss) در مجموعه داده های شرکت و Criteo کسب می کند.

	Company*		Criteo	
	AUC	LogLoss	AUC	LogLoss
LR	0.8640	0.02648	0.7686	0.47762
FM	0.8678	0.02633	0.7892	0.46077
FNN	0.8683	0.02629	0.7963	0.45738
IPNN	0.8664	0.02637	0.7972	0.45323
OPNN	0.8658	0.02641	0.7982	0.45256
PNN*	0.8672	0.02636	0.7987	0.45214
LR & DNN	0.8673	0.02634	0.7981	0.46772
FM & DNN	0.8661	0.02640	0.7850	0.45382
DeepFM	0.8715	0.02618	0.8007	0.45083

جدول ۲: عملکرد بر روی پیش بینی CTR

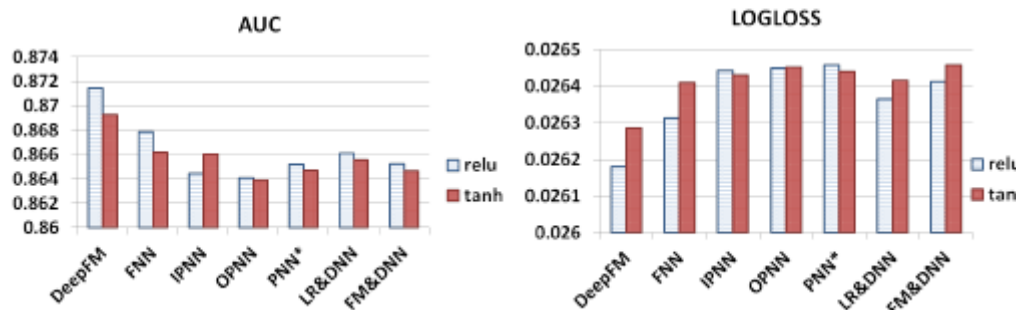
به طور کلی، مدل پیشنهادی DeepFM ما توانسته رقبای خود را به ترتیب با ارقام بیش از ۰.۳۷٪ و ۰.۴۲٪ از نظر AUC و Logloss در مجموعه داده های شرکت شکست است. در واقع، بهبود کمی در ارزیابی آنلاین AUC احتمالاً منجر به افزایش چشمگیری در CTR آنلاین خواهد شد. همانطور که در [چنگ و همکاران، ۲۰۱۶] گزارش شده است، در مقایسه با LR مدل Wide & Deep میزان AUC را ۰.۲۷۵٪ (آفلاین) بهبود می دهد و بهبود بخشی CTR آنلاین ۳.۹٪ است. گردش روزانه اپ استور شرکت میلیون ها دلار است، بنابراین حتی افزایش چند درصدی در CTR سالانه میلیون ها دلار اضافی را به ارمغان می آورد.

۳_۳ مطالعه با حداکثر پارامتر

ما تاثیر شیوه پارامترهای حداکثری از مدل های مختلف deep در مجموعه داده های شرکت را مطالعه می کنیم. ترتیب اینگونه است: (۱) فعال سازی توابع؛ (۲) نرخ رها کردن؛ (۳) تعداد اعصاب در هر لایه؛ (۴) تعداد لایه های پنهان؛ (۵) شکل شبکه.

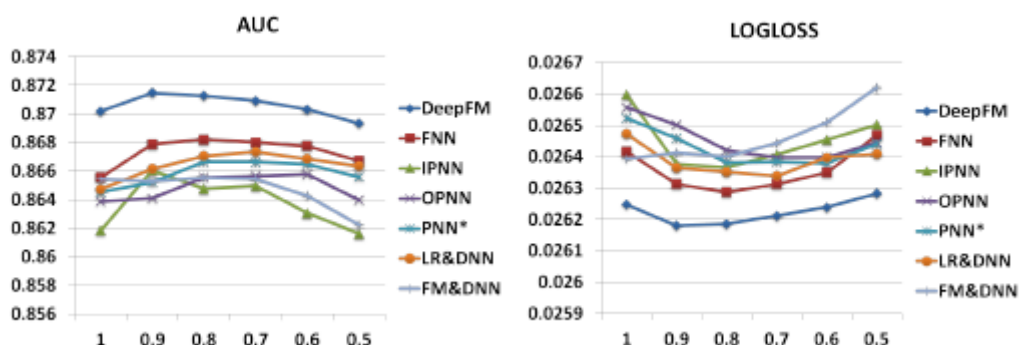
۱_۳_۳ تابع فعال سازی

با توجه به [کو و همکاران، ۲۰۱۶]، relu و tanh برای مدل‌های deep تر از sigmoid مناسب تر هستند. در این مقاله، عملکرد مدل های deep را در هنگام استفاده از relu و tanh مقایسه می کنیم. همانطور که در شکل ۷ نشان داده شده است، Relu مناسب تر از tanh برای تمام مدل های deep است، به جز IPNN. دلیل احتمالی این است که رولا پراکندگی را القا می کند.



۳-۲-۲ خروجی

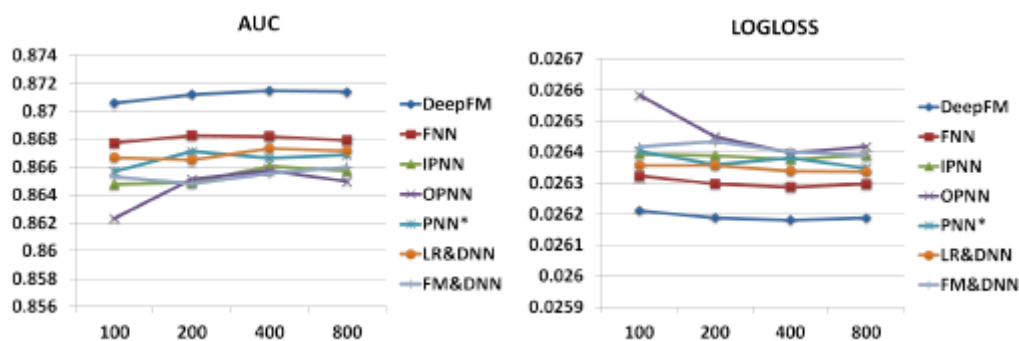
خروج [Srivastava et al., 2014] مربوط به احتمال اینکه یک نورون در شبکه نگهداری شود دارد. Dropout یک روش درست برای مقابله با دقت و پیچیدگی شبکه عصبی است. ما خروج را ۰,۸، ۰,۹، ۰,۷، ۰,۶، ۰,۵، تنظیم می کنیم. همانطور که در شکل ۸ نشان داده شده است، تمام مدل ها قادر به رسیدن به بهترین عملکرد خود در زمانی که خروج درست تنظیم شده است، هستند. (از ۰,۶ تا ۰,۹). نتیجه نشان می دهد که اضافه شدن تصادف منطقی به مدل می تواند ثبات مدل را تقویت کند.



شکل ۸: مقایسه AUC و Logloss از خروج.

۳-۳-۳ تعداد نورون ها در لایه

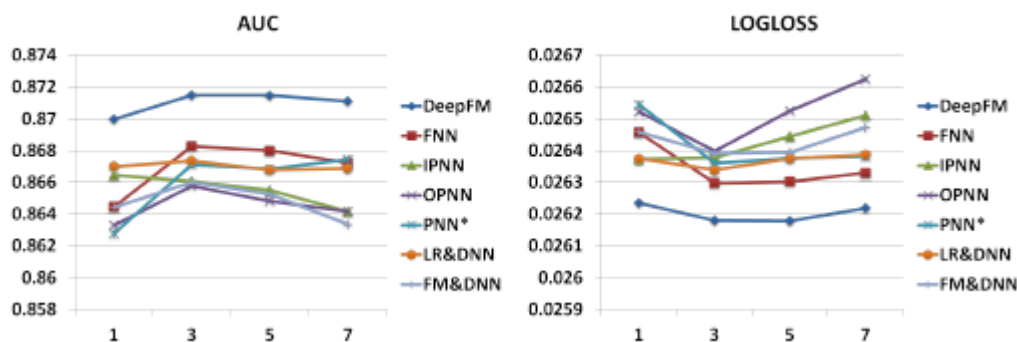
وقتی عوامل دیگری باقی می ماند، افزایش تعداد نورون ها در هر لایه پیچیدگی را به وجود می آورد. همانطور که در شکل ۹ می بینیم، افزایش تعداد نورون ها همیشه سودمند نیست. به عنوان مثال، DeepFM با هنگامی که تعداد نورون ها در هر لایه از ۴۰۰ تا ۸۰۰ افزایش می یابد؛ حتی بدتر از آن ثابت مواجه می شود، OPNN هنگامی که تعداد نورون ها از ۴۰۰ تا ۸۰۰ افزایش می یابد فعالیت بدتری خواهد داشتو این به این دلیل است که یک مدل پیچیده بیش از حد پیچیده است. در مجموعه داده های ما، ۲۰۰ یا ۴۰۰ نورون در هر لایه انتخاب خوبی است.



شکل ۹: مقایسه AUC و Logloss از تعداد نورون ها.

۳-۳-۴ تعداد لایه های مخفی

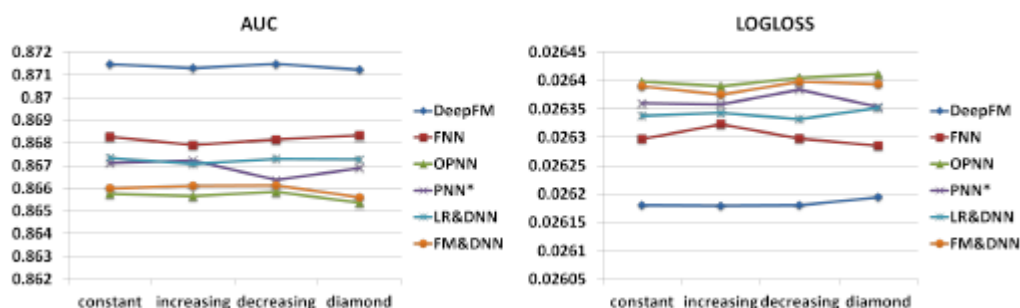
همانطور که در شکل ۱۰ نشان داده شده است، افزایش تعداد لایه های پنهان عملکرد مدل ها را در ابتدا بهبود می بخشد با این حال، اگر تعداد لایه های پنهان افزایش یابد، عملکرد آنها کاهش می یابد. این پدیده نیز به دلیل overfitting است.



شکل ۱۰: مقایسه AUC و Logloss از تعداد لایه ها.

۳-۳-۵ شکل شبکه

ما چهار شکل مختلف شبکه را آزمایش می کنیم: ثابت، افزایشی، کاهششی، و الماس. هنگامی که شکل شبکه را تغییر می دهیم، تعداد لایه های پنهان و تعداد کل نوروں ها را تعدیل می کنیم. به عنوان مثال، هنگامی که تعداد لایه های پنهان ۳ و تعداد کل نوروں ها ۶۰۰ است، چهار شکل متفاوت وجود دارد: ثابت (۲۰۰-۲۰۰)، افزایششی (۱۰۰-۲۰۰-۳۰۰)، کاهششی (۳۰۰-۲۰۰-۱۰۰)، و الماس (۱۵۰-۳۰۰-۱۵۰). همانطور که مشاهده می کنیم از شکل ۱۱، شکل "ثابت" شبکه تجربی بهتر از سه گزینه دیگر است، که مطابق با مطالعات قبلی است (لاروشل و همکاران، ۲۰۰۹).



شکل ۱۱: مقایسه AUC و Logloss از شکل شبکه.

۴. کار مرتبط

در این مقاله یک شبکه عصبی deep جدید برای پیش بینی CTR پیشنهاد شده است. مرتبط ترین دامنه های مرتبط پیش بینی CTR و یادگیری deep در سیستم پیشنهادی بودند. در این بخش، ما درباره کارهای مرتبط در این دو حوزه بحث می کنیم.

پیش بینی CTR نقش مهمی در سیستم توصیه کننده بازی می کند [Richardson et al., 2007; Juan et al., 2016; McMahan et al., 2013]. علاوه بر مدل های خطی تعمیم یافته و FM، چند مدل دیگر برای پیش بینی CTR پیشنهاد می شود، مانند مدل مبتنی بر درخت [He et al., 2014]، مدل مبتنی بر تنسور [رندل و اسمیت، Thieme، 2010]، پشتیبانی از بردار ماشین [چانگ و همکاران، ۲۰۱۰]، و مدل بلیسی [Graepel et al., 2010].

دامنه وابسته دیگر، یادگیری deep در سیستم های توصیه کننده است. در بخش ۱ و بخش ۲،۲، چندین مدل یادگیری deep پیش بینی CTR قبلاً ذکر شده است، بنابراین ما درباره آنها در اینجا بحث نمی کنیم. چندین مدل یادگیری deep در وظایف توصیه شده به غیر از پیش بینی CTR پیشنهاد شده است (به عنوان مثال [Covington et al., 2016; Salakhutdinov et al., 2007; van den Oord et al., 2013; Wu et al., 2016; Zheng et al., 2016; Wu et al., 2017; Zheng et al., 2017]. [Salakhutdinov et al., 2007; Sedhain et al., 2015; Wang et al., 2015]). پیشنهاد می کنند که فیلترینگ همگانی را از طریق یادگیری deep بهبود بخشد. نویسندگان [وانگ و وانگ، ۲۰۱۴؛ van den oord و همکاران، ۲۰۱۳] ویژگی های محتوا را از طریق یادگیری deep به منظور بهبود عملکرد توصیه های موسیقی استخراج می کنند. [چن و همکاران، ۲۰۱۶] یک شبکه یادگیری deep را برای در نظر گرفتن هر یک از ویژگی های تصویر و ویژگی های اساسی تبلیغات نمایش می دهد. [Covington et al., 2016] یک چارچوب یادگیری deep دو مرحله ای برای ویدئو توصیه YouTube توسعه می دهد.

۵. نتیجه گیری

در این مقاله، ما DeepFM را پیشنهاد می دهیم. DeepFM یک شبکه عصبی مبتنی بر ماشین فاکتورسازی برای پیش بینی CTR، برای غلبه بر کاستی های مدل های پیشرفته و عملکرد بهتر است. DeepFM یک جزء deep و یک مولفه FM به طور مشترک را آموزش می دهد. این بهبود عملکرد را از این مزایا به دست می آورد: (۱) هیچ پیش آموزشی لازم نیست؛ (۲) هر دو کنش های ویژگی high- و low-order یاد می گیرد؛ (۳) DeepFM یک استراتژی مشترک از ویژگی جاسازی شده برای جلوگیری از مهندسی ویژگی معرفی می کند. ما آزمایش های گسترده ای را در دو مجموعه داده های دیتا (مجموعه داده Criteo و مجموعه داده یک برنامه تجاری فروشگاه داده) برای مقایسه اثربخشی و کارایی DeepFM و مدل های پیشرفته تر انجام دادیم. نتایج آزمایش های ما نشان می دهد که (۱) DeepFM از نظر AUC و Logloss در هر دو مجموعه داده بهتر از مدل های به روز است؛ (۲) بهره وری DeepFM با مدل کارآمد deep تر در وضعیت پیشرفته قابل مقایسه است. در آینده برای مطالعاتی که صورت خواهد پذیرفت دو مورد جالب وجود دارد. اولین مورد این است که برخی از استراتژیها (مانند معرفی لایه های جمع آوری) را برای تقویت توانایی یادگیری مفاهیم کنش های ویژگی highorder مفید جستجو کنید. دیگر این است که DeepFM را بر روی یک خوشه GPU برای مشکلات بزرگ مقیاس امتحان کنید.

- **[1] Wide & Deep Learning for Recommender Systems**
Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah
Google Inc., submitted 2016
- **[2] DeepFM: A Factorization-Machine based Neural Network for CTR Prediction**
Huifeng Guo, Ruiming Tang², Yunming Ye, Zhenguo Li², Xiuqiang He²
¹Shenzhen Graduate School, Harbin Institute of Technology, China
²Noah's Ark Research Lab, Huawei, China, submitted 2017