دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر

پروژه تحقیقاتی درس شبکه‌های عصبی

# ماشین تورینگ عصبی

نگارش

علیرضا مازوچی

استاد درس

دکتر رضا صفابخش

تیر ۱۴۰۱


بسم الله الرحمن الرحيم

# چکیده

در این قسمت چکیده پایان نامه نوشته می‌شود. چکیده باید جامع و بیان‌کننده خلاصه‌ای از اقدامات انجام‌شده باشد.  در چکیده باید از ارجاع به مرجع و ذکر روابط ریاضی، بیان تاریخچه و تعریف مسئله خودداری شود.

## واژه‌های کلیدی:

کلیدواژه اول، ...، کلیدواژه پنجم (نوشتن سه تا پنج واژه کلیدی ضروری است)

# فهرست مطالب

عنوان صفحه

# فهرست اشکال

شکل صفحه

# فهرست جداول

جدول صفحه

# فصل اول

## مقدمه

# فصل دوم

# ماشین تورینگ عصبی

## ۲-۱ ساختار و محاسبات ماشین تورینگ عصبی

ماتع شامل دو جز است:

۱. کنترل‌گر شبکه که می‌تواند یک شبکه‌عصبی جلورو یا یک شبکه عصبی بازگشتی باشد.

۲. یک واحد حافظه خارجی که یک ماتریس حافظه $N * W$ است. $N$ تعداد واحدهای حافظه و $W$ ابعاد هر سلول حافظه را نمایش می‌دهد.[۱]

فارغ از آنکه کنترل‌گر بازگشتی باشد یا خیر، کل معماری بازگشتی محسوب می‌شود چراکه ماتریس حافظه در طول زمان نگهداری می‌شود. کنترل‌گر سرهای خوانده و نوشتن دارد که به ماتریس حافظه دسترسی دارد. تاثیر یک عمل خواندن یا نوشتن روی یک سلول حافظه خاص با مکانیسم توجه نرم[1] وزن‌دهی می‌شود. این مکانیسم آدرس‌دهی مشابه مکانیسم توجه استفاده‌شده در یادگیری ماشین عصبی است به جز آنکه آدرس‌دهی وابسته به موقعیت را با آدرس‌دهی وابسته به محتوای موجود در مکانیسم توجه را ترکیب می‌کند.[۱]

به طور خاص برای یک ماتع در هر گام زمانی $t$ برای هر سر خواندن و نوشتن کنترل‌گر یک تعدادی پارامتر را به عنوان خروجی می‌دهد. این پارامترها برای محاسبه وزن $w_t$ بر روی $N$ خانه حافظه در ماتریس حافظه $M_t$ استفاده می‌شوند. نحوه محاسبه $w_t$ در رابطه ۲-۱ آورده شده است.[۱]

$$w_t^c(i) \leftarrow \frac{exp(\beta_t K[k_t, M_t(i)])}{\sum_{j=0}^{N-1} exp(\beta_t K[k_t, M_t(j)])} \qquad (۱-۲)$$

در رابطه ۲-۱ برخی از پارامترها دارای محدودیت‌هایی هستند:[۱]

- $\beta_t \leq 0$
- $g_t \in [0, 1]$
- $\sum_k s_t(k) = 1$
- $\forall_k s_t(k) \leq 0$
- $\gamma_t \leq 1$

در رابطه ۲-۱ $w_t^c$ آدرس‌دهی وابسته به محتوا را فراهم می‌کند. $k_t$ یک کلید جستجو در حافظه را نشان می‌دهد و $K$ مطابق رابطه ۲-۲ یک معیار شباهت مانند شباهت کسینوسی است.[۱]

[1] Soft Attention Mechanism

$$K[u,v] = \frac{uv}{||u||.||v||} \quad (۲-۲)$$

با یک سری از محاسبات مطابق روابط ۲-۳، ۲-۴ و ۲-۵ ماتع‌ها امکان تکرار بر روی وزن‌های حافظه فعلی و قبلا محاسبه‌شده را خواهند داشت. رابطه ۲-۳ به شبکه اجازه می‌دهد تا بین بردار وزن قبلی یا فعلی انتخاب کند که از کدام استفاده کند. رابطه ۲-۴ امکان تکرار از طریق حافظه با عمل کانولوشن وزن فعلی و یک کرنل کانوولوشنی جابجایی[۲] یک بعدی را فراهم می‌کند. رابطه ۲-۵ رخداد تارشدن[۳] که به واسطه عمل کانوولوشن رخداده است را اصلاح می‌کند.[۱]

$$w_t^g \leftarrow g_t w_t^c + (1g_t) w_{t1} \quad (۳-۲)$$

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(ij) \quad (۴-۲)$$

$$w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_{j=0}^{N-1} \tilde{w}_t(j)^{\gamma_t}} \quad (۵-۲)$$

سپس بردار $r_t$ مطابق رابطه ۲-۶ به وسیله‌ی یک سر[۴] خواندن خاص در زمان $t$ محاسبه می‌گردد.[۱]

$$r_t \leftarrow \sum_{i=0}^{N-1} w_t(i) M_t(i) \quad (۶-۲)$$

نهایتا مطابق رابطه ۲-۷ و ۲-۸ هر سر نوشتن ماتریس حافظه را در گام $t$ با محاسبه بردارهای جانبی پاک‌کردن یعنی $e_t$ و جمع‌کردن یعنی $a_t$ تغییر می‌دهد.[۱]

[2] Shift

[3] Blurring

[4] Head

$$\tilde{M}_t(i) \leftarrow M_{t-1}(i)[1 - w_t(i)e_t] \quad (۷-۲)$$

$$M_t(i) \leftarrow \tilde{M}_t(i) + w_t(i)a_t \quad (۸-۲)$$

## ۲-۲ وظایف یادگیری ترتیبی

برای ماتع‌ها چندین وظیفه مصنوعی در نظر گرفته شده است که تمام آن‌ها از نوع مسئله یادگیری ترتیبی[۵] است؛ زمینه‌ای که آن‌ها در آن توانمند هستند. این وظایف عبارت اند از:

- رونوشت‌گیری[۶]: برای وظیفه رونوشت‌گیری یک دنباله تصادفی از بردارهای بیت با یک نماد خاص به عنوان پایان دنباله به شبکه داده می‌شود. این وظیفه نیاز دارد تا شبکه دنباله ورودی را نگه دارد و سپس آن را از حافظه برگرداند.

- رونوشت‌گیری تکرارشونده[۷]: مشابه وظیفه رونوشت‌گیری دنباله‌ای از بردارهای بیتی تصادفی به شبکه داده می‌شود. برخلاف وظیفه رونوشت‌گیری بعد از دنباله یک عدد که نشان دهنده تعداد دفعاتی است که باید دنباله در خروجی ظاهر شود به شبکه داده می‌شود.

- یادآوری انجمنی[۸]: این وظیفه نیز یک مسئله یادگیری دنباله با دنباله‌های متشکل از بردارهای بیتی تصادفی است. در این مورد ورودی به چندین عنصر تقسیم می‌شود که هر کدام شامل بردارهای ۳×۶ بعدی است. بعد از آنکه یک دنباله از آیتم‌ها و نماد پایانی دنباله به شبکه داده می‌شود. خروجی صحیح عنصر بعدی دنباله ورودی بعد عنصر کوئری است.[۱]

[5] Sequence Learning
[6] Copy
[7] Repeat Copy
[8] Associative Recall

# فصل سوم

# توسعه‌های شبکه

## ۳-۱ ماشین تورینگ عصبی تکاملی

تکامل عصبی توپولوژی‌های تقویت‌کننده[۱] که در ادامه آن را به اختصار تکتوت می‌نامیم با یک جمعیت[۲] از شبکه‌های عصبی ساده شروع می‌کند و سپس آن‌ها را در طی نسل‌ها[۳] با افزودن رئوس جدید و اتصالات به کمک جهش[۴] پیچیده‌تر می‌کند. با تکامل شبکه‌ها از این را لازم نیست توپولوژی شبکه‌ها از پیش دانسته شده باشد. تکتوت به طرز فزاینده‌ای در شبکه‌های پیچیده جستجو می‌کند تا یک سطح مناسب از پیچیدگی را پیدا کند. ویژگی مهم تکتوت این است که هم توپولوژی و هم وزن‌های یک شبکه را تکامل می‌دهد. چراکه به سادگی و تدریجی پیچیدگی را افزایش می‌دهد و این باعث می‌شود که یک شبکه مناسب با اندازه مینیمال حاصل شود.[۲]

بر پایه ماتع و با استفاده از تکتوت مدل ماشین تورینگ عصبی تکاملی[۵] که در ادامه آن را به طور اختصار ماتع تکاملی می‌نامیم معرفی شده است. در این روش توپولوژی و وزن‌های شبکه عصبی کنترل‌گر با کمک تکتوت یاد گرفته می‌شود. بنابراین برخلاف ماتع استاندارد نیاز به دانش پیشین[۶] نیست و شبکه می‌تواند باتوجه به پیچیدگی وظیفه رشد پیدا کند. ماتع تکاملی اغلب توپولوژی‌های فشرده برای حل یک وظیفه خاص پیدا می‌کند؛ در نتیجه جلوی جستجوی غیرضروری در فضای با ابعاد بالا گرفته می‌شود. به علاوه ماتع تکاملی قادر به جل مسائل یادگیری مستمر پیچیده است. چراکه شبکه از مشتق استفاده نمی‌کند و می‌تواند از توجه سخت[۷] و مکانیسم جابجایی[۸] استفاده کند که امکان تعمیم خوب برای دنباله‌های بلند در وظیفه رونوشت‌گیری را فراهم می‌کند. به علاوه یک نوار[۹] پویا و از نظر تئوری با اندازه بی‌نهایت قابل استفاده است.[۲]

ماتع تکاملی یک سر تکی ترکیبی خواندن/نوشتن دارد. این شبکه بردار نوشتن $w$ با اندازه $M$، ورودی کنترل درون‌یابی[۱۰] نوشتن $i$، ورودی کنترل پرش محتوا $j$ و سه ورودی کنترل جابجایی $s_l$ (جابجایی چپ)، $s_0$ (بدون جابجایی) و $s_r$ (جابجایی راست) را خروجی می‌دهد. اندازه بردار نوشتن $M$ اندازه هر خانه حافظه بر روی نوار را مشخص می‌کند. جزء درون‌یابی نوشتن امکان مخلوط‌کردن مقادیر فعلی نوار و بردار نوشتن در موقعیت نوشتن را فراهم می‌کند. $M_h(t)$ محتوای نوار در موقعیت سر فعلی $h$ در زمان $t$، $i_t$ درون‌یابی نوشتن و $w_t$ بردار نوشتن است. برای تمام این‌ها در زمان $t$ رابطه ۳-۱ را خواهیم

[1] Neuroevolution of Augmenting Topologies(NEAT)
[2] Population
[3] Generations
[4] Mutation
[5] Evolvable Neural Turing Machine
[6] Prior Knowledge
[7] Hard Attention
[8] Shift Mechanism
[9] Tape
[10] Interpolation

داشت.[۲]

$$M_h(t) = M_h(t1)(1i_t) + w_t i_t \tag{۱-۳}$$

پرش محتوا مشخص می‌کند که آیا سر باید به موقعیتی در حافظه حرکت کند که بیشترین شباهت را به بردار نوشتن دارد یا نه. یک پرش محتوا انجام می‌شود اگر مقدار ورودی کنترل از ۵.۰ بیشتر شود. شباهت بین بردار نوشتن $w$ و بردار حافظه $m$ مطابق با رابطه ۳-۲ حساب می‌شود.[۲]

$$s(w, m) = \frac{\sum_{i=1}^{M} |w_i m_i|}{M} \tag{۲-۳}$$

در گام زمانی $t$ اقدامات زیر به ترتیب انجام می‌شود:

۱. بردار نوشتن $w_t$ برای موقعیت فعلی $h$ بدست می‌آید. این بردار با محتوای موجود باتوجه به درون‌یابی نوشتن $i_t$ درون‌یابی می‌شود.

۲. اگر ورودی کنترل پرش محتوا $j_t$ بزرگ‌تر از ۵.۰ شود، سر به مکانی در نوار که بیشترین شباهت به بردار نوشتن $w_t$ دارد حرکت می‌کند.

۳. سر به یک موقعیت چپ‌تر، راست‌تر روی نوار حرکت می‌کند یا در همان جا ثابت می‌ماند که این وابسته به مقادیر ورودی کنترل جابجایی $s_l$، $s_0$ و $s_r$ است.

۴. مقادیر نوار را در موقعیت جدید سر می‌خواند و بر می‌گرداند.[۲]

## ۳-۲ ماشین تورینگ عصبی ابر تکاملی

در کدگذاری‌های مستقیم مانند تکتوت هر بخش از نمایش جواب به یک تکه کوچک از ساختار نهایی جواب نگاشت می‌شود. عیب مهم این روش آن است که بخش‌های مختلف راه‌حل که به یک‌دیگر شبیه هستند نیز باید کد شوند و جداگانه کشف شوند. این ابراد با کدگذاری غیرمستقیم تا حد زیادی قابل حل است؛ در کدگذاری غیرمستقیم راه‌حل به شکل فشرده توصیف می‌شود و حجم اطلاعات کدشده می‌تواند کاهش بیابد. در کدگذاری غیرمستقیم به دلیل آنکه یک راه‌حل به شکل الگویی از پارامترها و نه تمام پارامترها نمایش پیدا می‌کند قدرت‌مند است.[۲]

روش ابر تکامل عصبی توپولوژی‌های تقویت‌کننده[11] که در ادامه به اختصار آن را ابرتوت می‌نامیم

[11] Hyper Neuroevolution of Augmenting Topologies(HyperNEAT)

یک افزونه از تکتوت است. در این افزونه به جای کدگذاری غیرمستقیم از کدگذاری مستقیم استفاده می‌شود. در تکتوت از شبکه‌های عصبی معمولی استفاده می‌شود درحالی که در ابرتوت از شبکه‌های تولید الگوی ترکیبی[۱۲] که در ادامه آن را به اختصار شبکه تات می‌نامیم استفاده شده است. شبکه تات برای کدگذاری ترکیب توابع طراحی شده‌اند که هر تابع در ترکیب مرتبط با یک منظم‌سازی[۱۳] است.[۲]

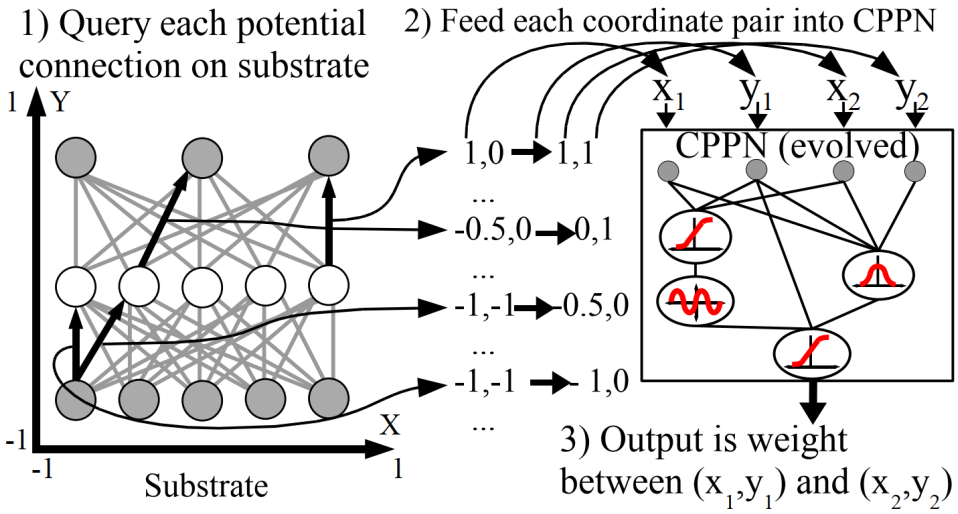
حسن شبکه تات آن است که به الگوهای مکانی اجازه می‌دهد که به عنوان شبکه‌هایی از توابع ساده نمایش پیدا کنند. این یعنی تکتوت می‌تواند با شبکه تات مانند شبکه‌های عصبی تکامل پیدا کند. شبکه‌های تات مشابه شبکه‌های عصبی هستند با این تفاوت که آن‌ها متکی بر بیشتر از یک تابع فعال‌سازی هستند. کدگذاری غیرمستقیم شبکه تات می‌تواند به طور فشرده الگوها با نظم‌هایی نظیر تقارن[۱۴]، تکرار[۱۵] و تکرار با تغییر[۱۶] را کد کنند. به عنوان مثال با انتخاب یک تابع گاوسین که خاصیت تقارن دارد الگوی خروجی نیز به سادگی متقارن خواهد شد. انتخاب یک تابع دوره‌ای مانند سینوس در حین تکرار قطعه‌سازی انجام می‌دهد. نهایتا تکرار با تغییر به سادگی با ترکیب یک تابع منظم (مانند سینوس یا گاوسین) با یک تابع نامنظم (مانند محور x نامتقارن) بدست می‌آید.[۲]

ایده اصلی ابرتوت آن است که شبکه‌های تات می‌تواند اتصال الگوها را کدکند. بدین طریق یک تکتوت می‌تواند یک شبکه تات که شبکه‌های عصبی بزرگ با منظم‌سازی‌ها و تقارن‌های خود نمایش می‌دهد را تکامل دهد.[۲]

عملکرد شبکه تات در تصویر ۳-۱ آورده شده است. شبکه‌های تات سنتی توابع هندسی هستند که الگوهای اتصال خروجی آن رئوسی در $n$ بعد است که $n$ تعداد ابعاد در فضای کارتزین است. یک شبکه تات که چهار ورودی با برچسب‌های $x_1$، $y_1$، $x_2$ و $y_2$ را دریافت کند و به عنوان خروجی مشخص می‌کند که اتصال بین نقاط دوبعدی $(x_1, y_1)$ و $(x_2, y_2)$ چه میزان است. بنابراین با داشتن یک شبکه تات آموزش یافته می‌توان با ارسال یک کوئری شامل هر دو راس در شبکه عصبی مقدار اتصال آن را بدست آورد و شبکه عصبی را ایجاد کرد.[۲]

همانطور که در توضیحات قبل بدان اشاره شد باید تمام گره‌های شبکه عصبی در یک بستر[۱۷] قرار داده شوند. یعنی مشخص شود که هر گره در شبکه عصبی در چه مختصاتی در فضا باید قرار بگیرد. مشابه چیزی که در سمت چپ تصویر ۳-۱ قابل مشاهده است. ایده اصلی در ماتع ابرتکاملی پیشنهاد یک بستر برای بهره‌مندی از شبکه تات در ماتع تکاملی بوده است.

[12] Compositional Pattern Producing Networks (CPPN)
[13] Regularity
[14] Symmetry
[15] Repetition
[16] Repetition with Variation
[17] Substrate




شکل ۳-۱: نحوه کارکرد شبکه تات[۲]

مجموعه‌ای از گره‌ها به مختصات بین -۱ تا +۱ در تمام ابعاد نظیر می‌شوند. (۱): هر اتصال ممکن در یک شبکه عصبی کوئری زده می‌شود تا مجاورت و وزن آن مشخص شود. خطوط جهت‌دار تیره نمایش‌داده شده در تصویر یک نمونه از اتصالاتی است که کوئری زده شده است. (۲): در درون یک شبکه تات یک گراف است که مشخص می‌کند کدام توابع فعال‌سازی به یکدیگر متصل هستند. همان‌طور که در شبکه عصبی اتصالات وزن‌دهی می‌شوند که خروجی یک تابع با چه وزنی به طرف دیگر اتصال برود، برای هر کوئری ارسال‌شده به شبکه تات جایگاه دو سر اتصال را به عنوان ورودی می‌گیرد و وزن اتصال را به عنوان خروجی می‌دهد. (۳): بنابراین شبکه تات می‌تواند الگوهای منظم از اتصالات در فضا را تولید کند.

در مدل ماتع ابرتکاملی شبکه تات نه تنها اتصال ورودی‌ها و خروجی‌های شبکه عصبی مرتبط با وظیفه را مشخص می‌کند بلکه اینکه اطلاعات آمده از حافظه چگونه باید در شبکه ادغام شوند و چگونه اطلاعات در حافظه نوشته شوند را هم مشخص می‌کند. زیرا ابرتوت که می‌تواند هندسه یک وظیفه را یاد بگیرد قائدتا باید بتواند الگوی هندسی اطلاعات خوانده شده از و نوشته شده در حافظه را نیز یاد بگیرد.[۲]

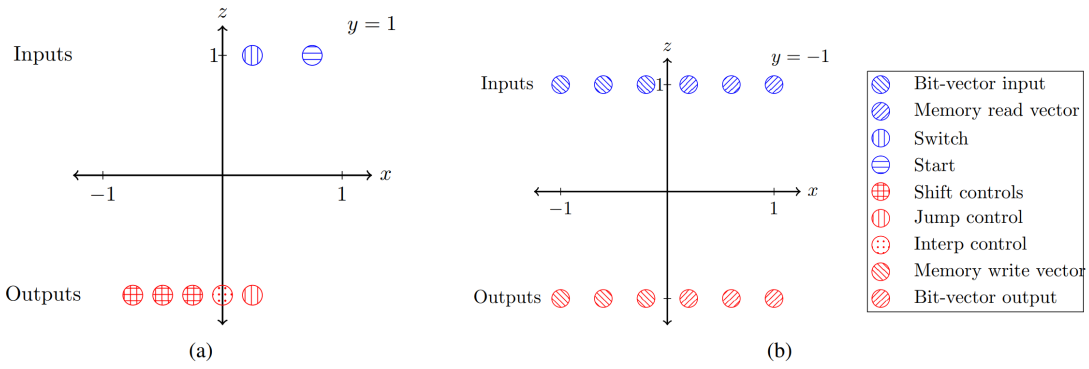
شبکه ماتع تکاملی ورودی‌های زیر را دارد:

- شروع: ورودی که هرگاه فعال می‌شود، ذخیره اعداد شروع می‌شود.
- تعویض: ورودی که هرگاه فعال شود، ذخیره اطلاعات خاتمه می‌یابد و شبکه باید مقادیر به خاطر سپرده‌شده را به یاد بیاورد.
- ورودی بردار بیتی: بردار بیتی که شبکه به عنوان ورودی می‌گیرد. توجه کنید قبل از آنکه ورودی تعویض فعال شود رنج این ورودی با بیت‌هایی که بعدا می‌خواند فعال می‌شود.
- ورودی خواندن حافظه: بردار حافظه که ماشین تورینگ در گام قبل خوانده است.[۲]

این شبکه خروجی‌های زیر را هم دارد:

- خروجی بردار بیتی: بردار بیتی که شبکه به عنوان خروجی می‌دهد. توجه کنید در حین دریافت ورودی این خروجی نادیده گرفته می‌شود.
- خروجی نوشتن حافظه: بردار حافظه‌ای که باید در حافظه نوشته شود.
- کنترل‌گرهای ماشین تورینگ: خروجی‌های کنترل مخصوص ماشین تورینگ یعنی پرش، درون‌یابی و سه کنترل جابجایی (چپ، راست و توقف)[۲]

در ادامه بستر طراحی‌شده برای وظیفه رونوشت‌گیری آورده می‌شود. این بستر در تصویر ۳-۲ نشان داده شده است. این بستر طراحی شده است که گره‌های ورودی بردار بیتی مختصات x را با گره‌های نوشتن بردار حافظه به اشتراک بگذارد و بالعکس با گره‌های خواندن بردار حافظه و گره‌های خروجی بردار بیتی. به علاوه ورودی تعویض مختصات x اش را با خروجی پرش به اشتراک می‌گذارد بنابراین شبکه را می‌تواند وادار به پرش به حافظه‌ای کند که خواندن را از آن شروع کرده است. در این مقاله اندازه بردارهای حافظه برابر با اندازه بردار بیتی است. به علاوه هیچ یک از بسترها شامل گره‌های مخفی مانند آن چیزی که نشان داده شده است و ممکن است مسائل با اندازه‌های بزرگ‌تر بدون گره مخفی را حل کند نیست.[۲]




شکل ۳-۲: بستر طراحی‌شده برای شبکه تات در مدل ماتع ابرتکاملی[۲]

تمام ورودی‌ها در $z = 1$ و تمام خروجی‌ها در $z = -1$ هستند. قسمت الف تمام گره‌ها در $y = 1$ را نشان می‌دهد که ورودی‌های شروع، تعویض و کنترل‌گرهای ماشین تورینگ هستند. لازم به ذکر است که مختصات $x$ برای ورودی تعویض و خروجی کنترل‌گر پرش یکسان است. قسمت ب گره‌ها را در $y = -1$ نشان می‌دهد که ورودی و خروجی‌های بردار حافظه و بردار بیتی را نشان می‌دهد. گره‌های ورودی بردار بیتی مختصات $x$ را با گره‌های نوشتن بردار حافظه به اشتراک می‌گذارد، درحالی‌که گره‌های خواندن بردار حافظه مختصات $x$ را با گره‌های خروجی بردار بیتی به اشتراک می‌گذارد.[۲]

در کنار خروجی شبکه تات که وزن هر اتصال را مشخص می‌کند، هر شبکه تات یک خروجی تابع

قدم[۱۸] اضافه دارد که خروجی بیان پیوند نامیده می‌شود. این خروجی مشخص می‌کند که آیا یک اتصال باید بیان شود یا خیر. اتصالات بالقوه برای هر ورودی در لایه‌های $y = 1$ و $y = -1$ به هر خروجی در لایه‌های $y = 1$ و $y = -1$ کوئری زده می‌شود. تعداد ورودی‌ها و خروجی‌ها در لایه $y = -1$ مطابق قسمت ب تصویر ۳-۲ وابسته به اندازه بردار بینی وظیفه رونوشت‌گیری است، که در مثال نشان‌داده شده اندازه بردار بیتی برابر با ۳ است. نورون‌ها به شکل یکنواخت در بازه‌های -۱ تا -۲.۰ در مختصات $x$ برای ورودی‌های بردار بیتی و بردار نوشتن حافظه استفاده می‌شود و بازه ۲.۰ تا ۱ برای بردار نوشتن حافظه و خروجی بردار بیتی استفاده می‌شود.[۲]

نهایتا برای تعیین میزان بایاس یک گره باید مختصات همان گره را هم به عنوان گره مبدا و هم به عنوان گره مقصد به شبکه تات کوئری زد.[۲]

## ۳-۳ ماشین تورینگ عصبی پویا

[18] Step Function

# فصل چهارم

# کاربردها

# فصل پنجم

# پیاده‌سازی و نتایج

# فصل ششم

# جمع‌بندی و نتیجه‌گیری

# منابع و مراجع

[1] Collier, Mark and Beel, Joeran. Implementing neural turing machines. In International Conference on Artificial Neural Networks, pages 94–104. Springer, 2018.

[2] Merrild, Jakob, Rasmussen, Mikkel Angaju, and Risi, Sebastian. Hyperentm: evolving scalable neural turing machines through hyperneat. In International Conference on the Applications of Evolutionary Computation, pages 750–766. Springer, 2018.