



هدف از این تمرین، انجام تحلیل مبتنی بر قید (Constraint-Based Analysis) بر روی شبکه‌های عصبی است؛ بدین معنا که قیود و روابط ریاضی حاکم بر شبکه در قالبی سازگار با حل‌گرهای SMT بازنویسی می‌شود تا از طریق استنتاج منطقی، صحت عملکرد شبکه در شرایط مختلف ورودی و امکان برقراری ویژگی‌های مطلوب آن (نظیر قرار گرفتن خروجی در محدوده تعریف شده) ارزیابی شود.

## فرمت ورودی:

در پوشه پیوست، تعدادی فایل yml وجود دارد که هر کدام تعریف یک شبکه عصبی کوچک می‌باشند. شرح خلاصه ساختار فایل در جدول شماره ۱ آمده است:

<b>model:</b>	هر فایل شامل دو قسمت اصلی model و property است. فعلا مدل فقط اطلاعات لایه‌ها را نمایش می‌دهد.
<b>layers:</b>  - type: "Dense"  units: 1  activation: "relu"  weights:  - [1.5, -1.0]  bias: [0.2]	لایه‌های شبکه هر کدام جداگانه ذیل layers تعریف می‌شوند. برای تمرین فعلی تمام لایه‌ها dense هستند در این مثال فقط یک لایه، شامل یک نورون وجود دارد که دو ورودی می‌گیرد و یک خروجی ارائه می‌دهد
<b>Property:</b>  inputs:  x1: [0, 1]  x2: [0, 1]  outputs:  y: ["<", 1.8]	در این قسمت محدودیت‌های روی ورودی و خروجی به صورت بازه تعریف می‌شود.

جدول شماره ۱: ساختار کلی فایل‌های yml.

برای سادگی:

- تمام مثالها فقط شامل لایه‌های dense هستند
- activation تمام نورون‌ها RELU در نظر گرفته شده است
- در تمام مثالها فقط یک خروجی وجود دارد (لایه آخر یک نورون دارد)

برای انجام این تمرین لازم است فایل `yaml` خوانده شده و ساختار شبکه به قالب مناسب برای `SMT-Solver` تبدیل شود. همچنین با تعریف متغیرهای حقیقی سمبولیک، قيود ورودی و خروجی مدل سازی شده و صحت آن تایید (یا رد) شود. همچنین:

- در صورت تایید شدن محدودیت خروجی، پیام مناسب چاپ شود و در صورت رد شدن، مثال نقض نمایش داده شود.
- زمان مورد استفاده برای استنتاج اندازه گیری و چاپ شود.

## پیوست ۱: فایل‌های تعریف شبکه

شش فایل تعریف شبکه در پوشه همراه تمرین موجود است:

Correct.yml

Incorrect.yml

One\_hidden.yml

Two\_hidden.yml

Three\_hidden.yml

CNN.yml

می‌توانید از دو مثال اول برای بررسی صحت کد خود استفاده کنید.

## پیوست ۲: بخش امتیازی

در مثال‌های ارائه شده در این تمرین، همه لایه‌ها از نوع `dense` بودند. آیا می‌توانید برای لایه‌های کانولوشن هم بررسی صحت قید خروجی را انجام دهید؟ یک شبکه عصبی کوچک شامل لایه کانولوشن در مثال `CNN.yml` قرار داده شده است که می‌توانید برای شروع و تست کد خود از آن استفاده کنید. در این مثال:

- ورودی یک تصویر  $4 \times 4$  پیکسل تک رنگ است که هر پیکسل مقداری بین صفر و یک دارد؛
- یک لایه کانولوشن با فیلتر  $2 \times 2$  و `ReLU` در ورودی قرار دارد که روی تصویر حرکت می‌کند. برای سادگی `stride` برابر یک و `padding` برابر صفر قرار داده شده و در نتیجه در خروجی یک تصویر  $3 \times 3$  تولید می‌شود؛
- لایه `Flatten`، خروجی قبلی را به یک بردار با طول ۹ تبدیل می‌کند.
- لایه آخر شامل یک نورون که ۹ ورودی را دریافت کرده و خروجی را تولید می‌کند.

### پیوست ۳: کار با SMT-Solver

در انتخاب SMT-Solver محدودیتی وجود ندارد. ولی باتوجه به حجم منابع آموزشی مرتبط، Z3 و CVC5 گزینه‌های مناسب‌تری می‌باشند. برای هر یک از این حل‌گرها آموزشهای بسیار زیاد و متنوعی اعم از ویدیو و کد در اینترنت موجود است ولی پس از شروع کار و برای رفع مشکلات حتما به اسناد مرجع اصلی که در لینک‌های زیر آمده مراجعه کنید:

<https://microsoft.github.io/z3guide/>

<https://cvc5.github.io/docs/cvc5-1.3.1/>

همچنین Z3 در ویندوز نصب می‌شود. ولی باتوجه به اینکه هردو حل‌گر، بومی Linux هستند، در صورتی که از سیستم‌عامل ویندوز استفاده می‌کنید WSL را فعال کنید. راهنمایی بیشتر در لینک زیر آمده است:

<https://learn.microsoft.com/en-us/windows/wsl/install>

## پیوست ۴: نحوه ارزیابی

ارزیابی تمرین براساس میزان صحت مدل‌سازی و اجرا و مستندسازی انجام می‌شود. همچنین برای هردوبخش (اصلی و امتیازی) لازم است محدودیت‌های کد خود را شرح داده و در صورتی که پیشنهادی برای رفع آن به نظرتان می‌رسد بیان کنید:

صحت مدل‌سازی شبکه و تبدیل آن به قیود منطقی (۳۰٪)

درستی اجرا با SMT-Solver (۳۰٪)

مستندسازی و نظم گزارش (یا کامنت‌های مناسب در کد) شامل شرح مراحل انجام کار، فرضیات بکار رفته، بیان محدودیت‌ها و... (۴۰٪)

بخش امتیازی (۲۰٪)

زمان تحویل تمرین:

- ارسال فایل نهایی: ۳۰ آبان ۱۴۰۴

- ارائه حضوری: ۳ آذر ۱۴۰۴